Universitat Politècnica de Catalunya
Programa de Doctorat de Ingeniería Civil

Departament de Matemàtica Aplicada III

Adaptive hybrid discontinuous methods for fluid and wave problems

by

Giorgio Giorgiani

Doctoral Thesis

Advisors: Antonio Huerta and Sonia Fernández-Méndez

Barcelona, April 2013

Abstract

Adaptive hybrid discontinuous methods for fluid and wave problems

Giorgio Giorgiani

This PhD thesis proposes a p-adaptive technique for the Hybridizable Discontinuous Galerkin method (HDG).

The HDG method is a novel discontinuous Galerkin method (DG) with interesting characteristics. While retaining all the advantages of the common DG methods, such as the inherent stabilization and the local conservation properties, HDG allows to reduce the coupled degrees of freedom of the problem to those of an approximation of the solution defined only on the faces of the mesh. Moreover, the convergence properties of the HDG solution allow to perform an element-by-element postprocess resulting in a superconvergent solution.

Due to the discontinuous character of the approximation in HDG, p-variable computations are easily implemented. In this work the superconvergent postprocess is used to define a reliable and computationally cheap error estimator, that is used to drive an automatic adaptive process. The polynomial degree in each element is automatically adjusted aiming at obtaining a uniform error distribution below a user defined tolerance. Since no topological modification of the discretization is involved, fast adaptations of the mesh are obtained.

First, the p-adaptive HDG is applied to the solution of wave problems. In particular, the Mild Slope equation is used to model the problem of sea wave propagation is coastal areas and harbors. The HDG method is compared with the continuous Galerkin (CG) finite element method, which is nowadays the common method used in the engineering practice for this kind of applications. Numerical experiments reveal that the efficiency of HDG is close to CG for uniform degree computations, clearly out-

performing other DG methods such as the Compact Discontinuous Galerkin method. When p-adaptivity is considered, an important saving in computational cost is shown.

Then, the methodology is applied to the solution of the incompressible Navier-Stokes equations for the simulation of laminar flows. Both steady state and transient applications are considered. Various numerical experiments are presented, in 2D and 3D, including academic examples and more challenging applications of engineering interest. Despite the simplicity and low cost of the error estimator, high efficiency is exhibited for analytical examples. Moreover, even though the adaptive technique is based on an error estimate for just the velocity field, high accuracy is attained for all variables, with sharp resolution of the key features of the flow and accurate evaluation of the fluid-dynamic forces. In particular, high degrees are automatically located along boundary layers, reducing the need for highly distorted elements in the computational mesh. Numerical tests show an important reduction in computational cost, compared to uniform degree computations, for both steady and unsteady computations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Discontinuous Galerkin methods (DG) are finite element methods that are locally conservative and stable, and allow to achieve high-order accuracy. The DG formulation uses discontinuous approximation element by element, with the information that passes through the elements by means of numerical fluxes. Since their introduction, DG methods have been used to solve a large variety of partial differential equations, gaining an increasing interest in the scientific computing community. In fact, they have proved to be suited for the construction of robust high-order numerical schemes on arbitrary unstructured and non-conforming grids, for a variety physical fenomena. The main drawback of DG methods, compared to continuous Galerkin (CG), is the increased number of degrees of freedom (DOF) due to the duplication of the nodes belonging to faces of the elements. It has been often claimed that this disadvantage becomes less important for high-order elements, due to the increasing number of interior nodes. However, until very recently, DG methods coupled the nodes in the interior of the elements, for high-order approximations, with neighboring element nodes, even for reduced stencil approaches such as interior penalty methods, see for example Arnold (1982), and Compact Discontinuous Galerkin method (CDG), see Peraire and Persson (2008). This coupling precluded any technique, such as *static condensation* of the interior nodes for CG, to reduce the number of DOF. Under these circumstances, DG are penalized respect to CG.

However, recently a numerical technique called hybridization is earning interest among the DG users. Similarly to static condensation, hybridization of DG methods allows to reduce the globally coupled degrees of freedom to those of an approximation of the solution defined only on the boundaries of the elements (trace solution). The hybridizable discontinuous Galerkin method (HDG), was formally introduced by Cockburn et al. (2009) for second order elliptic problems, and then extended to the solution of a great variety of physical fenomena, see for example Nguyen et al. (2009, 2011c,a,b, 2010) and Peraire et al. (2010). The similarity between hybridization in DG and static condensation in CG is highlighted by Kirby et al. (2011), both techniques leading to an important reduction in the number of DOF for high-order computations. Nevertheless, Arnold and Brezzi (1985) showed that hybridization is not only an implementation expedient to reduce the linear system size, but the trace unknown contains extra informations on the exact solution. In fact, unlike other DG methods and standard CG, HDG exhibits optimal convergence in $\mathcal{L}^2$ norm not only for the primal unknown of the problem but also for its derivative, opening the path to an element-by-element postprocess which provides a superconvergent solution.

Hence, high-order HDG can to obtain high accuracy with very efficient computations. Obviously, the most efficient way to minimize the computational effort is to adapt the discretization in each area of the domain to satisfy a requested error tolerance. Adaptivity is the natural way for refining the mesh only in the zones where more accuracy is needed, either for a goal oriented computation or for achieving a uniform error distribution. DG methods provide a natural framework for implementing adaptive techniques: due to the discontinuous character of the approximation in each element, non-matching grids containing hanging nodes and non-uniform approximation degrees can easily be handled.

The most typical adaptive techniques are based on modifying the element size (h-adaptivity, being h the element size) or modifying the polynomial degree of the element (p-adaptivity, being p the polynomial degree of the approximation). Considering a single element, local h-refinement is performed generating successive nested elements of the same degree, while p-refinement is performed replacing the element with an-

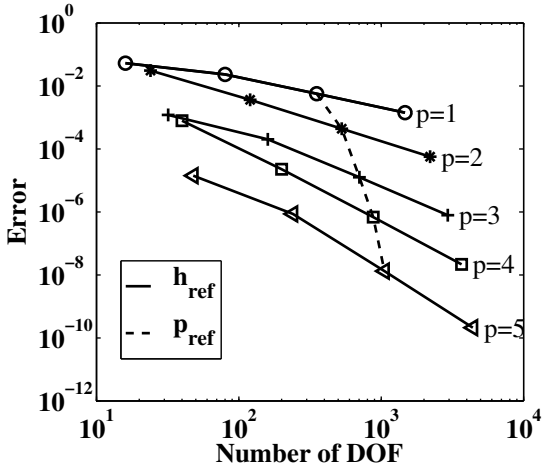


Figure 1.1: h-refinement vs p-refinement.

other of increased degree. Despite the existing vast literature on adaptive algorithms, p-adaptive techniques have not received the attention they deserve, probably due to its cumbersome implementation in the context of continuous approximations. Nevertheless, degree adaptive algorithms, usually in combination with mesh size adaptation, have proved to be very efficient, specially in the context of discontinuous approximations and smooth solutions. In fact, if the exact solution of the problem is smooth, the convergence of p-refinement is faster than the one of h-refinement, see for example Babuška and Szabo (1991). As an example, in Figure 1.1 is shown a comparison between h and p refinement for a second order elliptic problem. The markers on the continuous curves refers to four different element sizes, while the dashed line joints points of equal h and increasing p.

This example underlines the fact that increasing the polynomial degree provides faster convergence to the exact solution. For this reason, p-adaptivity is preferred, whenever is possible, to h-adaptivity. Moreover, no topological modification of the discretization is involved in p-adaptivity, thus the invariance of the elemental connectivity typically leads to a simpler implementation. However, it is well known that increasing the polynomial degree of the approximation in areas of the domain where the exact solution presents discontinuities, will only introduce more oscillations in the approximate solution and may result in loss of stability. Hence, the a priori rule of

thumb for *hp*-adaptive techniques is that *h*-refinement can be used to isolate singularities while *p*-enrichment, away from singularities, delivers exponential convergence, see for example Eskilsson (2011) and Remacle et al. (2003). A survey of the adaptive methods proposed in the past in the framework of DG is done in Section 1.2.1.

One of the ingredients to implement an adaptive method is an *a-posteriori* error estimator to drive the automatic adaptive process. The error estimation is, in most cases, an expensive computation. In fact, it usually requires the evaluation of an *improved* solution that entails a non-negligible over-cost. Some of the most common error estimators used in the literature are summarized in Section 1.2.2.

This thesis focuses on a *p*-adaptive technique for the HDG method. Hence, no problems with discontinuities are considered. The adaptive technique allows to automatically modify the polynomial degree of the approximation independently in each element. Thus, the efficiency of the computation is drastically improved, since the computational complexity is calibrated to obtain the desired tolerance in each part of the domain. Moreover, the adaptive technique also simplifies the initial design of the mesh, reducing the need to a priori adjust the discretization to the finite element solution.

The adaptive algorithm is driven by an error estimator derived exploiting the superconvergent properties of HDG and involving only elemental computations. Thus, the error estimation is an inexpensive computation compared to the solution, as will be shown in the examples. High reliability of the error estimator is also found.

The *p*-adaptive HDG is used for the solution of incompressible flow problems and wave propagation problems. Both steady-state and transient solutions are considered. High efficiency of the computations and high accuracy is shown for both physical problems considered.

In the following section the main contributions and partial objectives of the thesis are summarized.

1.1 Objectives and overview

The goal of this thesis is to propose a p-adaptive HDG method providing high-order accuracy and high computational efficiency. The proposed method will be applied to wave propagation problems and incompressible viscous flow problems. The main contributions of the thesis are:

1. **p-adaptive HDG for wave problems** The application to the problem of sea wave propagation in harbors is considered in Section 3.2. Large domains and small geometrical features are the principal sources of computational complexity in this problem. Moreover, the sloping bottom generates a non uniform wavelength that requires different spatial resolutions in different areas of the propagation domain. One of the main issues in wave problems is the ability of correctly resolve the wave with a sufficient number of nodes. The proposed adaptive technique allows to automatically capture the oscillations placing the correct polynomial degree in each element, to obtain a uniform error distribution below a user defined tolerance. High efficiency computations are obtained with the proposed algorithm. The adaptive technique is described in Section 3.1, while a comparison with high-order CG computations is proposed in Section 3.2. Results of this research can be found also in Giorgiani et al. (2013b).

2. **p-adaptive HDG for fluid problems** Incompressible flows represent another challenging application where adaptivity is crucial. In fact, sharp gradients of the solution are often embedded in uniform streams, and the solution can change in time, requiring to repeatedly update the discretization to capture the evolution of the flow. Section 3.3 is devoted to the application of the p-adaptive technique to the Navier-Stokes equations. Several examples are presented in 2D and 3D, including a test with analytical solution that allows to evaluate the efficiency of the error estimator, and more challenging applications of engineering interest such as a vertical axis wind turbine. High efficiency of the computations is shown also in this case. Results of this research can be found also in Giorgiani et al. (2013a).

3. **Comparison for p-uniform computations** Another critical issue in wave propagation is the problem of the dispersion, that is, the difference between the wavelength of the finite element solution and the exact solution. It is well known that high-order approximations reduce this problem. DG methods can also represent an alternative for reducing the dispersion error. In Section 2.2, a thorough comparison between high-order and low-order elements is performed in a 2D scattering problem. In this framework, the computational efficiency of HDG is compared with the one of CG and CDG. The comparison criteria are based on the dimension of the final linear system and on the total computational time to obtain the solution. Results of this comparison can also be found in Giorgiani et al. (2013).

To develop the contributions of the thesis, the following partial objectives are also considered:

- **Derivation and implementation of different finite element methods for the solution of the problem of wave propagation in harbors.** The prediction of the wave height in coastal areas is fundamental for harbor design. The Mild Slope equation (MSE), derived by Berkhoff (1972), is taken as model for the simulation of this phenomenon. In Appendix F is detailed the derivation of the MSE starting from the equation of motion of an incompressible inviscid fluid. Both frequency and time-frequency formulations are considered. A DG method with upwind fluxes is derived and implemented for the time-frequency formulation of the MSE. Details of this analysis are presented in Appendix E.

  Resulting clearly more efficient the frequency analysis of the MSE respect to the time-frequency analysis –mainly due to the fact that it is an intrinsically 2D model– the latter is discarded and the work focuses on the frequency formulation of the MSE. Hence, three high-order finite element methods are derived for the MSE in frequency formulation, that is CG, CDG and HDG. The weak forms of the three methods are presented in Section 2.2. Appendix A presents a convergence study of HDG applied to the MSE varying the stabilization parameter τ.

For HDG, details on the implementation can also be found in Appendix B.

- **Implementation of high-order HDG for the incompressible Navier-Stokes equations.** The HDG method is implemented for the Navier-Stokes equations in 2D and 3D. The weak form of the problem is presented in Section 2.3. A comparison of the computational efficiency between low-order and high-order elements is presented, using as test case the evaluation of the aerodynamic characteristics of a NACA 0012 airfoil. Details of the implementation can be found in Appendix C.

- **Derivation of HDG fractional step methods for solving 3D fluid problems.** When solving the Navier-Stokes equations in 3D, the solution of the linear system generated by the finite element discretization becomes critical. In fact, using a direct solver is computationally inefficient. The use of an iterative solver is also cumbersome due to the saddle-point structure of the linear system matrix. An alternative approach is proposed in Appendix D, where two fractional steps methods are presented for the time integration of the unsteady Navier-Stokes equations with HDG. The fractional step methods allow to circumvent the problem represented by the saddle-point matrix produced by the constrained system, splitting the advancing in time in two simpler non-constrained problems.

1.2 State of the art

The importance of adaptive simulations has been pointed out by various authors since the introduction of the finite element method (FEM), see for example Babuška and Rheinboldt (1978, 1979); Wu et al. (1990); Prudhomme and Oden (1999); Remacle et al. (2003, 2006); Oñate et al. (2006). On one hand, for wave propagation problems, areas where concentration of reflections occur often coexist with zones of unperturbed propagation of the wave. Moreover, the wavelength can vary from area to area, leading to the necessity of different spatial resolution to correctly capture the oscillations in the whole computational mesh. On the other hand, for fluid problems, the motion

of an incompressible fluid subject to different type of boundary conditions and body forces often presents zones with sharp gradients enclosed in areas where the solution is almost uniform. Furthermore, these features can change in time and move in the domain, leading to the necessity to repeatedly update the mesh to be able to correctly capture the evolution of the fluid motion. In both cases, the refinement pattern is almost unpredictable a priori, and the efficiency of the automatic adaptive algorithm is crucial.

In the following section, a survey of the adaptive techniques proposed in the literature is done.

1.2.1 Discontinuous Galerkin adaptive methods

The first adaptive techniques in FEM are based on the works of Babuška and Rheinboldt (1978, 1979) and the successive elaborations by Gago et al. (1983), Löhner et al. (1985) and Zienkiewicz and Zhu (1987) among others, which developed both h and p refinements to solve elasticity and flow problems, in the context of continuous FEM. Nowadays, adaptivity in CG is widely used, even though the implementation can become cumbersome. If hanging nodes are introduced with local h and p refinement, special techniques are required to impose the C_0 continuity between the elements, such as special constrains on the hanging points or transition elements. Non-conforming CG methods have successfully been implemented through the use of mortar elements, see Kim et al. (2005), or the use of interface matching conditions, see Fischer et al. (2002).

On the other hand, for DG methods, the discontinuous character of the solution provides a natural framework for adaptive techniques. Introduced by Reed and Hill (1973) as a technique to solve neutron transport problems, the DG method has become popular more than two decades later as a method for solving fluid dynamics or electromagnetic problems. Beyond advantages for adaptivity, DG methods provide a natural stabilization to the solution due to the inter-element fluxes, and are very suited for parallelization. Bassi and Rebay (1997) introduced a DG method for the

Navier-Stokes equations and then Cockburn and Shu (1998) introduced the so-called local discontinuous Galerkin (LDG) method, generalizing the original DG method of Bassi and Rebay.

In the 1990s, with the increasing interest in DG, the first adaptive techniques were introduced in the method. Bey (1994), and then Bey and Oden (1996), introduced a *hp*-adaptive technique in DG for hyperbolic conservation laws. The mesh modification was driven by a local (element-wise) a posteriori error estimator computed with elemental problems on an increased polynomial space (because of the Galerkin orthogonality, see Section 1.2.2). The choice between h or p refinement in each element was based on the smoothness of the solution in the element. This criterion has always been used in successive *hp*-techniques such as, for example, the one proposed by Houston and Süli (2001) for first-order hyperbolic problems. In this case, the local residual terms were multiplied with weights computed using a dual problem, and a linear functional was used as quantity of interest. This idea was then extended to nonlinear hyperbolic conservation laws by Hartmann and Houston (2002a) and then to the compressible stationary Euler equations by Hartmann and Houston (2002b), but only with h-adaptivity. Hartmann (2006) extended the h-adaptive technique with the duality approach to the compressible Navier-Stokes equation, while Wang and Mavriplis (2009) solved the same problem with a *hp*-technique using an element-wise indicator for capturing the shock.

In the field of wave propagation, the first works on adaptive FEM are due to Stewart and Hughes (1996, 1997) and Bouillard and Ihlenburg (1999), which developed h-adaptive techniques for CG in acoustics. However, few adaptive DG applications have been proposed so far for the solution of wave problems, see Remacle et al. (2006) for *hp*-adaptive DG and Kubatko et al. (2009) and Michoski et al. (2011b) for p-adaptive DG, for the solution of the shallow-water equations.

Transient computations of unstable flows provide an application where adaptivity is crucial. Remacle et al. (2003) used adaptive DG with orthogonal basis to solve transient compressible flow problems such as Rayleigh-Taylor instabilities and shocks. An error indicator based on the inter-element jumps of density was used, that is, no

postprocess of the solution was used to estimate the error. This entails a fast estimation crucial for repeated mesh adaptation. Burbeau and Sagaut (2005) proposed a p-adaptive technique for viscous flows with shocks. The key ingredient was a discontinuity sensor devoted to the detection of the shock. Then, in regions where the solution was quasi-uniform and in the vicinity of shocks the degree of the polynomial basis was decreased, while a high-degree basis was used in regions of smooth fluctuations of the flow. A similar approach was also used by Michoski et al. (2011a), where a family of slope limiters was proposed for p-adaptive solutions of advection-reaction equations.

More recent publications in the field of adaptive DG focus on the solution of large 3D applications of computational fluid dynamics (CFD), in particular turbulent flows and RANS equations, see for example Hartmann et al. (2010, 2011), but only with h-adaptivity.

Adaptivity of hybrid DG methods is an open field. Hybridization of DG methods derives from the mixed methods of Raviart and Thomas when, as Arnold and Brezzi (1985) indicate, the continuity constrain was eliminated from the finite element space and imposed by means of Lagrange multipliers on the inter-element boundaries. The idea was then developed by Cockburn and Gopalakrishnan (2005) and Cockburn et al. (2009), were the HDG method was formally developed for elliptic problems. Nowadays, the method has been applied to typical problems, such as elasticity, electromagnetism and fluids. The first adaptive HDG method has been proposed in Giorgiani et al. (2013b) for wave propagation problems.

One crucial ingredient in any adaptive technique is the error estimator devoted to drive the adaptive process. In the following section are reviewed the techniques to estimate the error in adaptive algorithms.

1.2.2 Error estimators for adaptive computations

Error estimation is crucial for deciding where to refine/coarse the discretization. Since the late 1970s, several strategies have been developed to estimate the discretization

error of the finite element solution, see Huerta et al. (1999) for a general discussion. Basically, two types of error estimation procedure are available. The so called a-priori error estimation provides information on the asymptotic behavior of the error but it is not designed for giving informations on the actual error on a given mesh. In contrast, a-posteriori error estimation is a technique that involves postprocessing the numerical solution itself in such a way that important and useful informations can be obtained on the actual error present in the numerical solution. Ainsworth and Oden (2000) and Gratsch and Bathe (2005) review various a-posteriori error estimators.

The basic characteristics that an efficient error estimator should have for adaptive algorithms, are:

- the error estimate should be accurate in the sense that the predicted error is close to the actual (unknown) error;

- the error estimator should be computable and simple, with the error estimate inexpensive to compute when compared on the total runtime of the analysis.

The output of the error estimator is an elemental error index E_i which represents the contribution of the element Ω_i to the global error of the computation. Two broad classed of methods for computing the a-posteriori elemental error index E_i are:

- recovery methods, based on the fact that the gradient of the finite element solution is in general discontinuous across the inter-element boundaries. Thus, the idea is to postprocess the gradient and find an estimate of the error by comparing the postprocessed gradient and the non postprocessed gradient of the approximation;

- residual methods, where the error estimation derives from the approximate evaluation of the residual of the original equation: they can further be divided into explicit schemes, which employ directly the residual of the current approximation, and implicit schemes, which use the residuals indirectly and generally involve the solution of a small linear algebraic system.

Recovery methods have been successfully used in CG adaptive techniques, see for example the works of Zienkiewicz and Zhu (1987) and Zienkiewicz et al. (1999). However, this kind of estimators present some drawbacks, see Ainsworth and Oden (2000) and Prudhomme and Oden (2003) for details, and have never been used for DG methods.

In DG, residual methods have always been preferred. Residual methods use the equation for the residual of the equation to obtain informations on the discretization error. Given the variational formulation of a generic non-linear problem,

$$\mathcal{B}(u, v) = \mathcal{F}(v), \quad \forall v \in V, \tag{1.1}$$

being V the functional space of admissible solutions, the finite element problem reads: find $u_h \in V_h$ such that

$$\mathcal{B}(u_h, v) = \mathcal{F}(v), \quad \forall v \in V_h, \tag{1.2}$$

where V_h is the finite element space. Defining the error as $e_h = u - u_h$, the equation for the residual is

$$\mathcal{B}(u_h + e_h, v) - \mathcal{B}(u_h, v) = \mathcal{R}(v), \quad \forall v \in V \tag{1.3}$$

where $\mathcal{R}(v) = \mathcal{F}(v) - \mathcal{B}(u_h, v)$. The solution of (1.3) must be sought in an increased space with respect to the solution space, because of the Galerkin orthogonality,

$$\mathcal{R}(v) = 0, \quad \forall v \in V_h,$$

thus a direct solution of (1.3) in the whole domain is more expensive than the original problem (1.2). Nevertheless, the approach of solving Equation (1.3) on an increased space has been used for adaptive techniques. For example Lacasse et al. (2007) proposed a DG adaptive technique for advection-reaction problems, where the residual equation was solved using quadratic interpolation functions, while linear functions were used for the solution of the original problem.

However, this procedure is quite unusual, and residual type error estimators seek

for an approximation of (1.3) which is less expensive than the original problem at hand. Explicit methods, for example, use available data from the finite element solution and do not require solving any auxiliary problem. They are also called *error indicators*, since the output is not a sharp error estimation, but contains useful information for driving the adaptive process. They are fast and computationally cheap, thus particularly suited for repeated estimations and mesh adaptations typical of fast transient simulation. Remacle et al. (2003) used an explicit error estimator to drive the *hp*-adaptive DG method for solving compressible flow problems. Inter-element jumps of the density were used as error indicator. This is particularly effective in refining the discontinuities arising in compressible flows. Remacle et al. (2006) used the adaptive technique to solve the shallow-water equations. In this case, an error indicator based on the second derivative of the water height was used in smooth regions, while the gradient of the water height was used across discontinuities. Also for the shallow-water equations, Kubatko et al. (2009) proposed an error indicator based on the approximate gradient of the solution inside the element. Thus, explicit estimators are widely and effectively used. However, the derivation of such estimators rely on multiplicative constants which are unknowns, hence the error cannot be accurately estimate. Other approaches have also been proposed for time marching simulation requiring fast error approximations. For example Eskilsson (2011) proposed to use the solution computed on a p-1 mesh to evaluate the error of the p-mesh: this is equivalent to approximate the error of the p mesh with the error of the p-1 mesh. Again, an estimation of the error distribution among the elements is cheaply obtained, but the absolute estimation of the elemental error is poor.

On the other hand, implicit estimators retain the structure of the original equation as far as possible, defining a local boundary value problem with the residual as a data. Dependence on unknown constants is reduced by the solution process itself, see for example Parés et al. (2006, 2008, 2009). This kind of estimators have been used for adaptive DG by Bey and Oden (1996), which used the elemental residual method to obtain lower and upper bounds of the elemental error, solving a local boundary problem on an increased polynomial space.

Recently, new methods based on duality techniques have been introduced in the field of a posteriori error estimation. For adaptive DG, weighted residuals were used in the works of Houston and Süli (2001); Hartmann and Houston (2002a,b); Hartmann et al. (2010, 2011), with the weights computed from a dual problem solved on an improved mesh. The dual problem approach leads to optimal meshes with respect to an output of interest represented as a functional that can be either linear or nonlinear. For example, for the solution of steady compressible Navier-Stokes equations, Hartmann et al. (2010) used the dual problem for an efficient approximation of the aerodynamic force coefficients. However, the solution of the dual problem on an increased space can be more costly than the original problem.

An equivalent way to compute the residual on an increased space is to calculate an improved solution u_h^* using a finer mesh or a higher degree of interpolation. The comparison of the improved solution with the non-improved one provides obviously informations on the elemental error. Again, the computation of the improved solution can be a prohibitively expensive computation, if obtained with the same approach of the normal solution. In HDG, an improved solution can be obtained exploiting the superconvergent properties of the method, with a cheap element-by-element computation. This is the approach that has been followed in this thesis.

Chapter 2

High-order hybridizable discontinuous Galerkin method

In this chapter the HDG method is introduced for second order elliptic problems, and then the method is used to solve wave and fluid problems using high-order approximations.

The derivation of the method and the superconvergent postprocess are discussed in Section 2.1. In Section 2.2, HDG is used to solve the Mild Slope equation modeling sea wave propagation in harbors. High efficiency of high-order computations is shown. High-order elements, in this case $p > 2$, provide better accuracy than low-order elements for the same computational cost, or require less computational cost for a desired precision, even for engineering accuracy. HDG is also compared with CG, which is the standard method used to solve this kind of problem in engineering practice, and with CDG, that is, a classic DG method for second order elliptic problems. HDG outperforms CDG, and reveals to be an efficient alternative to CG, even without exploiting the possibility of p-adaptive computations.

In Section 2.3, HDG is used to solve the Navier-Stokes equations for incompressible flows. High-order elements again reveal to be a good choice to obtain high accuracy reducing the computational cost.

2.1 The HDG method for second order elliptic problems

The hybridization of a finite element method is based on the introduction of a new approximated solution which is defined only on the faces (in 2D and 3D) of the finite element mesh. As shown in the unified framework for hybrid methods introduced by Cockburn et al. (2009), several FE methods are hybridizable. In fact, *static condensation* of the interior nodes in high-order CG (see for example Zienkiewicz et al. (2005)) can be seen has a particular case of hybridization, see also Kirby et al. (2011). However, differently from static condensation, hybridization is not only an implementation trick to reduce the linear system size, since the trace variable contains *extra* informations on the solution that can be used to enhance the convergence of the method, see Arnold and Brezzi (1985) for more details. In the following, the derivation of HDG is detailed for the Laplace equation.

Consider the second order elliptic problem with Dirichlet boundary conditions in the open bounded domain $\Omega \subset \mathbb{R}^d$, with boundary $\partial\Omega_D$ and d the number of spatial dimensions

$$\begin{aligned} -\boldsymbol{\nabla} \cdot \boldsymbol{\nabla} u &= f \quad \text{in } \Omega, \\ u &= g_D \quad \text{on } \partial\Omega_D, \end{aligned} \tag{2.1}$$

where $f \in \mathcal{L}^2(\Omega)$. Suppose that Ω is partitioned in $\mathtt{n_{el}}$ disjoint subdomains Ω_i with boundaries $\partial\Omega_i$. The following formal definitions and notation are used for the computation domain and its broken counterpart

$$\overline{\Omega} = \bigcup_{i=1}^{\mathtt{n_{el}}} \overline{\Omega}_i,\ \Omega_i \cap \Omega_j = \emptyset \text{ for } i \neq j, \text{ and } \widehat{\Omega} := \bigcup_{i=1}^{\mathtt{n_{el}}} \Omega_i,$$

whereas the union of all interior faces is

$$\Gamma := \bigcup_{\substack{i,j=1 \\ i \neq j}}^{\mathtt{n_{el}}} \overline{\Omega}_i \cap \overline{\Omega}_j = \Big[\bigcup_{i=1}^{\mathtt{n_{el}}} \partial\Omega_i\Big] \backslash \partial\Omega,$$

and the union of all $\mathtt{n_{fa}}$ faces, Γ_i for $i = 1, \ldots, \mathtt{n_{fa}}$, is $\Gamma^+ := \Gamma \cup \partial\Omega$. Both Γ and Γ^+ have, if needed, their broken counterparts denoted, respectively, by $\widehat{\Gamma}$ and $\widehat{\Gamma}^+$.

The previous discretization allows to define the finite dimensional spaces: $\widehat{\mathcal{V}}_h$ and $\widehat{\Lambda}_h$, namely

$$\widehat{\mathcal{V}}_h := \left\{ v \in \mathcal{L}^2(\Omega) : v|_{\Omega_i} \in \mathcal{P}^{p_{\Omega_i}}(\Omega_i), \text{ for } i = 1, \ldots, \mathtt{n_{el}} \right\}, \tag{2.2a}$$

$$\widehat{\Lambda}_h := \left\{ \mu \in \mathcal{L}^2(\Gamma^+) : \mu|_{\Gamma_i} \in \mathcal{P}^{p_{\Gamma_i}}(\Gamma_i), \text{ for } i = 1, \ldots, \mathtt{n_{fa}} \right\}, \tag{2.2b}$$

where $\mathcal{P}^p$ denotes the space of polynomials of degree $\leq p$, while p_{Ω_i} and p_{Γ_i} are the polynomial degrees in element Ω_i and face Γ_i respectively. The space associated to the Dirichlet boundary is defined as $\widehat{\Lambda}_h(g_D) = \{\mu \in \widehat{\Lambda}_h : \mu = \mathbb{P}_\partial g_D \text{ on } \partial\Omega\}$, where $\mathbb{P}_\partial$ is the $L^2(\partial\Omega_D)$ projection into the space $\{\mu|_{\partial\Omega}, \forall \mu \in \widehat{\Lambda}_h\}$.

Remark 1. *In general, the polynomial degree for elements, and faces, can vary from element to element, and from face to face. More precisely, in all computations, given a map of elemental degrees, the interpolation degree p_{Γ_k} for a face shared by two elements, $\Gamma_k = \partial\Omega_i \cap \partial\Omega_j$, is set as the maximum value of the degree in Ω_i and Ω_j, that is $p_{\Gamma_k} = \max\{p_{\Omega_i}, p_{\Omega_j}\}$, and $p_{\Gamma_k} = p_{\Omega_i}$ when $\Gamma_k \in \partial\Omega_i \cap \partial\Omega$. This procedure ensures that for any element Ω_i the degree on all its faces is at least p_{Ω_i}, providing the desired accuracy in the element, see also Chen and Cockburn (2012a,b).*

Remark 2. *Standard nodal basis are used to represent these spaces. In Figure 2.1 the nodes corresponding to a polynomial degree $p = 5$ in a triangle are shown for the spaces introduced. Fekete node distributions (for straight-sided and curved elements) are considered to minimize ill-conditioning, see Taylor et al. (2000).*

Due to the discontinuities between elements, the *jump* $[\![\cdot]\!]$ and the *mean* $\{\cdot\}$ operators are defined along the interface Γ using values from the elements to the left and right of the interface (say, Ω_i and Ω_j), namely

$$[\![\circledcirc]\!] = \circledcirc_i + \circledcirc_j, \qquad \{\circledcirc\} = (\circledcirc_i + \circledcirc_j)/2.$$

The major difference between the mean and the jump operator is that the latter always

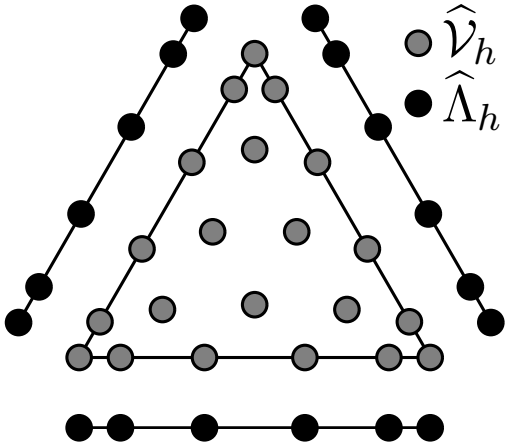


Figure 2.1: Nodes representing the spaces $\widehat{\mathcal{V}}_h$ and $\widehat{\Lambda}_h$.

involves the normal to the interface, see Montlaur et al. (2008) for more details.

Finally, the following scalar products are introduced, $\big(\cdot,\cdot\big)_D$ and $\big\langle\cdot,\cdot\big\rangle_B$, denoting respectively the $\mathcal{L}^2$ scalar product in any domain D and the $\mathcal{L}^2$ scalar product of the traces over B

The HDG formulation requires first to rewrite (2.1) as a system of first order equations taking into account the discontinuities of the approximation spaces between elements. Thus Equations (2.1) become:

$$\boldsymbol{q} + \boldsymbol{\nabla} u = 0, \qquad \text{in } \widehat{\Omega}, \tag{2.3a}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{q} = f, \qquad \text{in } \widehat{\Omega}, \tag{2.3b}$$

$$u = g_D, \qquad \text{in } \widehat{\Omega} \cap \Omega_D, \tag{2.3c}$$

$$[\![\boldsymbol{n}\cdot\boldsymbol{q}]\!] = 0, \qquad \text{on } \Gamma, \tag{2.3d}$$

$$[\![u\boldsymbol{n}]\!] = \boldsymbol{0}, \qquad \text{on } \Gamma, \tag{2.3e}$$

where $\boldsymbol{n}$ is the unitary outward normal vector. Calling $\boldsymbol{q}_h \in [\widehat{\mathcal{V}}_h]^d$ and $u_h \in \widehat{\mathcal{V}}_h$ the FE approximations of the unknowns, the weak form of the problem (2.3) becomes

$$\left.\begin{aligned} (\boldsymbol{q}_h, \boldsymbol{v})_{\Omega_i} - (u_h, \boldsymbol{\nabla}\cdot\boldsymbol{v})_{\Omega_i} + \langle \hat{u}_h, \boldsymbol{v}\cdot\boldsymbol{n}\rangle_{\partial\Omega_i} = 0, \\ -(\boldsymbol{q}_h, \boldsymbol{\nabla} v)_{\Omega_i} + \langle \hat{\boldsymbol{q}}_h\cdot\boldsymbol{n}, v\rangle_{\partial\Omega_i} = (f, v)_{\Omega_i}, \end{aligned}\right\} \text{ for } i = 1, \ldots, \mathtt{n_{el}}, \tag{2.4}$$

for all the test functions $(\boldsymbol{v}, v) \in [\widehat{\mathcal{V}}_h]^d \times \widehat{\mathcal{V}}_h$. In (2.4), $\hat{u}_h$ and $\hat{\boldsymbol{q}}_h$ are the numerical traces of the unknowns u and $\boldsymbol{q}$ on Γ^+. In HDG, the numerical trace $\hat{\boldsymbol{q}}_h$ is defined as

$$\hat{\boldsymbol{q}}_h = \boldsymbol{q}_h + \tau(u_h - \hat{u}_h)\boldsymbol{n}, \tag{2.5}$$

while $\hat{u}_h$ is set to $\mathbb{P}\partial g_D$ on $\partial\Omega_D$ and it is left as a new unknown of the problem on $\widehat{\Gamma}$.

The parameter τ is a positive stabilization parameter which is fundamental for the stability and accuracy of HDG. A detailed study of the possible choices can be found in Cockburn et al. (2008, 2009) and in Kirby et al. (2011). The optimal choice for τ will be discussed for wave and fluid problems in the relative sections. A study on the influence of this parameter on the convergence and superconvergence properties of HDG is also presented in Appendix A.

Remark 3. *The general form of the numerical traces introduced by Castillo et al. (2000) for a generic DG method is*

$$\hat{\boldsymbol{q}}_h = \{\boldsymbol{q}_h\} + C_{11}[\![u_h\boldsymbol{n}]\!] - \boldsymbol{C}_{12}[\![\boldsymbol{q}_h \cdot \boldsymbol{n}]\!], \tag{2.6a}$$

$$\hat{u}_h = \{u_h\} + \boldsymbol{C}_{12} \cdot [\![u_h\boldsymbol{n}]\!] + C_{22}[\![\boldsymbol{q}_h \cdot \boldsymbol{n}]\!]. \tag{2.6b}$$

The stability of the DG method is proved for any non-negative value of C_{11} and C_{22}. However, the accuracy of the method depends on the actual choice for these two parameters. For implementation issues, usually C_{22} is set to zero, allowing to solve for the auxiliary variable $\boldsymbol{q}_h$ in terms of u_h and recover a single equation in the unknown u_h. This approach is followed, for example, in the LDG method, see Cockburn and Shu (1998), and in CDG, see Peraire and Persson (2008) for details. However, DG methods with zero C_{22} parameter lead to suboptimal convergence in $\mathcal{L}^2$ norm for the $\boldsymbol{q}_h$ variable. In contrast, for HDG, it can be shown that the flux definition (2.5) *is equivalent to setting the coefficients in* (2.6) *as*

$$C_{11} = \frac{\tau_i\tau_j}{\tau_i + \tau_j}, \quad \boldsymbol{C}_{12} = \frac{1}{2}\frac{[\![\tau]\!]}{\tau_i + \tau_j}, \quad C_{22} = \frac{1}{\tau_i + \tau_j}, \tag{2.7}$$

that is, the coefficient C_{22} is different from zero for any finite value of τ. This provides,

for uniform p computations, optimal converging approximations with slope $p+1$ in $\mathcal{L}^2$ norm both for u_h and $\boldsymbol{q}_h$.

The strong point of the hybridization technique relies on the introduction of the new variable $\hat{u}_h \in \widehat{\Lambda}_h$, single valued for each face of the mesh. The unknown $\hat{u}_h$ is determined by the *jump condition* (2.3d), which in weak form reads

$$\sum_{i=1}^{\mathtt{n_{el}}} \langle \hat{\boldsymbol{q}}_h \cdot \boldsymbol{n}, \mu \rangle_{\partial\Omega_i} = 0,$$

leading to the final form of the HDG method for the Laplace equation: find $(\boldsymbol{q}_h, u_h, \hat{u}_h) \in [\widehat{\mathcal{V}}_h]^d \times \widehat{\mathcal{V}}_h \times \widehat{\Lambda}_h(g_D)$ such that

$$\left.\begin{aligned} (\boldsymbol{q}_h, \boldsymbol{v})_{\Omega_i} - (u_h, \boldsymbol{\nabla}\cdot\boldsymbol{v})_{\Omega_i} + \langle \hat{u}_h, \boldsymbol{v}\cdot\boldsymbol{n}\rangle_{\partial\Omega_i} &= 0, \\ (\boldsymbol{\nabla}\cdot\boldsymbol{q}_h, v)_{\Omega_i} + \langle \tau(u_h - \hat{u}_h), v\rangle_{\partial\Omega_i} &= (f, v)_{\Omega_i}, \end{aligned}\right\} \text{ for } i = 1, \ldots, \mathtt{n_{el}}, \tag{2.8a}$$

$$\sum_{i=1}^{\mathtt{n_{el}}} \langle \boldsymbol{q}_h \cdot \boldsymbol{n} + \tau(u_h - \hat{u}_h), \mu \rangle_{\partial\Omega_i} = 0, \tag{2.8b}$$

for all $(\boldsymbol{v}, v, \mu) \in [\widehat{\mathcal{V}}_h]^d \times \widehat{\mathcal{V}}_h \times \widehat{\Lambda}_h(0)$. It is worth noting that (2.8a) is a local system in each element Ω_i which does not involve unknowns of neighboring elements. Thus, (2.8a) can be solved element-by-element to express $\boldsymbol{q}_h$ and u_h as functions of $\hat{u}_h$. And consequently, Equation (2.8b) is the only equation coupling globally the variables of different elements. Replacing $\boldsymbol{q}_h$ and u_h, solution of (2.8a) in terms of $\hat{u}_h$, in (2.8b) yields a global system on the whole mesh skeleton for the trace variable, $\hat{u}_h \in \widehat{\Lambda}_h$. Once the global system is solved, $\boldsymbol{q}_h$ and u_h can be recovered for each element Ω_i using (2.8a).

HDG postprocessed solution

With the solution of the global problem described by (2.8b), the $\mathtt{n_{el}}$ local problems (2.8a) give an approximation of $(u_h, \boldsymbol{q}_h) \in \widehat{\mathcal{V}}_h \times [\widehat{\mathcal{V}}_h]^d$. Then, for each element, Equation (2.3a) can be solved using the corresponding Neumann boundary conditions and

restriction for solvability, namely

$$\left.\begin{aligned} -\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}u_h^* &= \boldsymbol{\nabla}\cdot\boldsymbol{q}_h && \text{in } \Omega_i \\ -\boldsymbol{n}\cdot\boldsymbol{\nabla}u_h^* &= \boldsymbol{n}\cdot\boldsymbol{q}_h && \text{on } \partial\Omega_i \\ \int_{\Omega_i} u_h^*\, d\Omega &= \int_{\Omega_i} u_h\, d\Omega \end{aligned}\right\} \text{ for } i = 1,\ldots,\mathtt{n_{el}}.$$

This induces a weak problem in a richer finite dimensional space, that is, find $u_h^* \in \widehat{\mathcal{V}}_h^*$ such that

$$\big(\boldsymbol{\nabla}u_h^*, \boldsymbol{\nabla}v\big)_{\Omega_i} = \big(\boldsymbol{q}_h, \boldsymbol{\nabla}v\big)_{\Omega_i} \text{ and } \big(u_h^*, 1\big)_{\Omega_i} = \big(u_h, 1\big)_{\Omega_i},$$

for all $v \in \widehat{\mathcal{V}}_h^*$ and $i = 1,\ldots,\mathtt{n_{el}}$, where $\widehat{\mathcal{V}}_h^*$ must be a bigger space than $\widehat{\mathcal{V}}_h$. In fact, with one degree more in the element-by-element polynomial approximation, i.e. $\widehat{\mathcal{V}}_h^* = \big\{v \in \mathcal{L}^2(\Omega) : v|_{\Omega_i} \in \mathcal{P}^{p_{\Omega_i}+1}(\Omega_i), \text{ for } i = 1,\ldots,\mathtt{n_{el}}\big\}$, it is sufficient to prove that u_h^* converges asymptotically at a rate of $p+2$ in the $\mathcal{L}^2$ norm for uniform p distribution, see Cockburn et al. (2008). In case of non-uniform distributions of p_{Ω_i}, superconvergence of the p-variable HDG postprocess solution, under the condition in Remark 1 is proved in Chen and Cockburn (2012a,b).

2.2 High-order HDG for wave propagation problems

Many engineering applications and physical phenomena, such as acoustic waves, electromagnetism and vibrations, are modeled with wave equations. The assumption of harmonic character of the solution leads to Helmholtz-type wave equations, with non-constant coefficients in the general case. Solving this problem for high frequencies leads to numerical difficulties because of the loss of the elliptic character and the oscillatory behavior of the solution. When tackling the problem with any finite element method, the characteristic size, h, of the spatial discretization has to be sufficiently small. Often, it is suggested to use constant kh, where k is the wavenumber, see Thompson and Pinsky (1994) and Ihlenburg and Babuška (1995b). In practice, this

means using a fixed number of nodes per wavelength (i.e. wave resolution) in each spatial direction. However, Ihlenburg and Babuška (1995b) showed that this rule guarantees an accurate solution only for low wavenumbers. In fact, Ihlenburg and Babuška (1997) also proved that the *a priori* error estimate in energy semi-norm of the finite element solution can be written as

$$e_h \leq C_1 \left(\frac{kh}{2p}\right)^p + C_2 k \left(\frac{kh}{2p}\right)^{2p}, \tag{2.10}$$

where C_1 and C_2 are two constants independent of k, h and p. The first term in (2.10) corresponds to the interpolation error. The second term is the pollution error, which is strictly related with the difference in wavelength between the exact and the finite element solution, that is, the dispersion error, see Steffens and Díez (2009) and Steffens et al. (2011) for more details. For large wavenumbers, the pollution error becomes the dominating term in (2.10) and it is responsible for the degradation of the finite element solution at a fixed wave resolution. Thus, Equation (2.10) shows that increasing the wavenumber requires increasing the number of nodes per wavelength to keep a fixed level of accuracy.

Many techniques have been proposed in the past to reduce the dispersion error. Stabilized finite elements is probably the most popular, see Chang (1990); Harari and Hughes (1992); Babuška and Sauter (2000); Harari and Magoulès (2004); Harari and Gosteev (2007). Embedding the oscillatory behavior of the solution in the approximating functions is another option to improve accuracy, see for instance Melenk and Babuška (1996); Cessenat and Despres (1998); Lacroix et al. (2003); Farhat et al. (2003); Strouboulis and Hidajat (2006); Gabard (2007); Gabard et al. (2011). Another alternative, obvious from (2.10), is to increase the order of the approximation p, see Ihlenburg and Babuška (1995a). This also coincides with recent results showing that high-order computations can be more efficient than low-order ones, see Vos et al. (2010); Cantwell et al. (2011); Huerta et al. (2012); Modesto et al. (2012); Giorgiani et al. (2012); Sevilla et al. (2013).

Discontinuous Galerkin methods can also be an alternative in this area, see for

example Feng and Wu (2009) and Alvarez et al. (2006). Feng and Wu (2009) showed that interior penalty discontinuous Galerkin method performs better than CG in the pre-asymptotic range (large element size) for the Helmholtz equation. Alvarez et al. (2006) proposed a discontinuous finite element formulation of the Helmholtz equation, which requires less DOF than CG for properly chosen parameters. Note however that Feng and Wu (2009) and Alvarez et al. (2006) are restricted to linear approximations and, moreover, CG is hampered by the fact that no stabilization is used. To the author knowledge there are no comparisons for high-order approximations between continuous and discontinuous Galerkin where pollution errors do not require to stabilize CG.

In what follows, HDG is compared to CG and to CDG in a 2D scattering problem for a wide range of polynomial degrees. The CDG method is taken as reference of classic DG methods for elliptic problems because it introduces the smallest stencil, as interior penalty, see Arnold (1982).

The comparison is carried out through various numerical examples. Two measures of the computational cost are considered: the dimension of the final linear system, and the total runtime. While the criterion based on the linear system dimension is implementation-free and gives also a measure of the hardware requirements to obtain the solution, it is not a completely fair comparison because it does not take into account the structure of the linear system matrix, which can be significantly different for different methods and different polynomial interpolations. Moreover, the assembly cost, which is also obviously different for the different cases, is not taken into account. On the other hand, the runtime criterion takes into account all the costs, but obviously the results may depend on the actual implementation used for each method. To minimize this dependency, the same code optimization has been used for all methods, see Roca et al. (2011) for details, and the same direct linear system solver has been used for all the computations. Note that this can be disadvantageous for DG methods because the structure of the information (constant bandwidth of block dense matrices) allows important gains in today's hardware, see for instance Kirby et al. (2011), and they are not exploited here.

2.2.1 Problem statement

The following form of the Helmholtz equation is considered

$$\boldsymbol{\nabla} \cdot (\beta \boldsymbol{\nabla} u) + k^2 \beta u = \hat{f}, \tag{2.11}$$

in an unbounded 2D domain where $\beta(x, y) \in \mathbb{R}$ is a material parameter, $k(x, y) \in \mathbb{R}$ is the wavenumber, and $\hat{f}$ is an harmonic source. This equation models wave propagation for several problems of academic and engineering interest. One of the most challenging ones is the propagation of sea waves with a slowly-varying bottom depth. In this case, the coefficients are non-constant, large domains with small geometrical features induce very large systems of equations, two significant digits of accuracy are sought, and the number of runs can be very large due to the wide range of input data. Equation (2.11) is in this case also known as the Berkhoff or Mild Slope equation, see Appendix F.

Boundary conditions in this case are, on one hand, for reflecting/absorbing boundaries, $\Gamma_{\mathtt{R}}$,

$$\boldsymbol{n} \cdot \beta \boldsymbol{\nabla} u - Ik\alpha\beta\, u = \hat{g} \text{ on } \Gamma_{\mathtt{R}},$$

where $I = \sqrt{-1}$ is the imaginary unit, $\hat{g}$ is a data function and $\alpha \in [0, 1]$ is a real coefficient. This coefficient is equal to zero on totally reflecting boundaries and to one on perfectly absorbing boundaries, see for instance Berkhoff (1976). On the other hand, unbounded scattering problems require the so-called Sommerfeld radiation condition

$$\lim_{r\to\infty} \sqrt{r}\Big(\frac{\partial}{\partial r} - Ik\Big)(u - u_0) = 0, \tag{2.12}$$

where r is the radial direction and u_0 the incident wave. The Sommerfeld radiation condition requires, in practice, the introduction of an artificial boundary and its corresponding boundary condition.

In this case it is usual to define a bounded computational domain and to introduce an artificial boundary, see among others Givoli (1992) and Givoli and Neta (2003). Note that the coefficient β is assumed constant at the artificial boundary and beyond, at least in the normal direction. Here, a Perfectly Matched Layer (PML) surrounds the

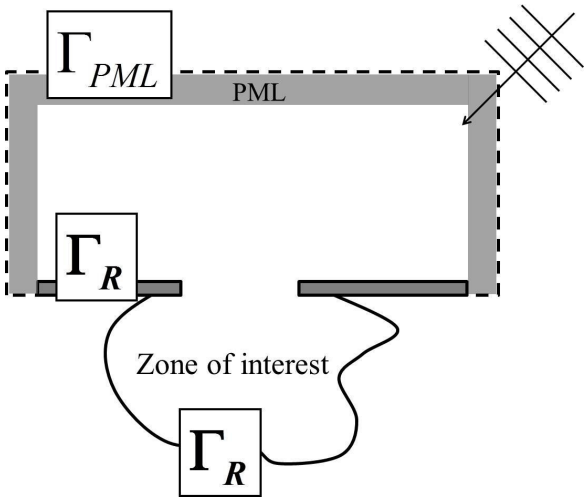


Figure 2.2: Problem statement.

computational domain in order to absorb outgoing waves, see for instance Berenger (1994); Abarbanel and Gottlieb (1998); Turkel and Yefet (1998); Bermúdez et al. (2004). The setup of the problem is illustrated in Figure 2.2, showing the PML region. The problem to be solved is then

$$\boldsymbol{\nabla} \cdot (\beta\, \mathbf{P}\, \boldsymbol{\nabla} u) + k^2 s_x s_y \beta\, u = s_x s_y f \qquad \text{in } \Omega, \tag{2.13a}$$

$$\boldsymbol{n} \cdot \beta \boldsymbol{\nabla} u - Ik\alpha\beta\, u = g \qquad \text{on } \Gamma_{\mathtt{R}}, \tag{2.13b}$$

$$\boldsymbol{n} \cdot \beta \mathbf{P} \boldsymbol{\nabla} u - Ik\beta\, u = 0 \qquad \text{on } \Gamma_{\mathtt{PML}}, \tag{2.13c}$$

where u is now the scattered field, Ω is the bounded computational domain including the PML, f and g are modifications of $\hat{f}$ and $\hat{g}$ accounting for the incident wave, $\mathbf{P}$ is the diagonal anisotropy matrix defining the absorption in the PML medium, namely

$$\mathbf{P} = \begin{pmatrix} s_y/s_x & 0 \\ 0 & s_x/s_y \end{pmatrix},$$

and s_x and s_y are the absorption parameters in the two Cartesian directions. Note that $s_x = s_y = 1$ outside the PML region. More details on the application of the PML to the Helmholtz equation can be found in Bermúdez et al. (2004). Equation

(2.13c) is a first order non-reflecting boundary condition discretizing (2.12) on $\Gamma_{\texttt{PML}}$, to minimize non-physical reflection from the PML outer boundary. Thus, $\partial\Omega = \overline{\Gamma}_{\texttt{R}} \cup \overline{\Gamma}_{\texttt{PML}}$ with $\Gamma_{\texttt{R}} \cap \Gamma_{\texttt{PML}} = \emptyset$, and no Dirichlet boundary conditions are imposed.

2.2.2 Finite element formulations

In this section the CG, CDG and HDG finite element formulations are recalled for the solution of Equation (2.13). Since the problem considered is 2D, in the following $d = 2$ is set.

The continuous Galerkin formulation

Introducing the FE space $\mathcal{V}_h$ of continuous functions in Ω, i.e.

$$\mathcal{V}_h := \left\{ v \in \mathcal{H}^1(\Omega) : v|_{\Omega_i} \in \mathcal{P}^p(\Omega_i), \text{ for } i = 1, \ldots, \texttt{n}_{\texttt{el}} \right\},$$

the standard discrete CG problem approximating (2.13) requires to find $u_h \in \mathcal{V}_h$ such that

$$a(u_h, v) = l(v) \quad \forall v \in \mathcal{V}_h, \tag{2.14}$$

where the bilinear and linear forms, $a(\cdot,\cdot)$ and $l(\cdot)$ are given by

$$a(u,v) = \big(\beta \mathbf{P} \boldsymbol{\nabla} u, \boldsymbol{\nabla} v\big)_{\Omega} - \big(\gamma u, v\big)_{\Omega} - I\big\langle k\alpha\beta u, v\big\rangle_{\Gamma_{\texttt{R}}} - I\big\langle k\beta u, v\big\rangle_{\Gamma_{\texttt{PML}}},$$
$$l(v) = \big\langle g, v\big\rangle_{\Gamma_{\texttt{R}}} - \big(s_x s_y f, v\big)_{\Omega},$$

with $\gamma := k^2 s_x s_y \beta$.

Note that high-order CG is always implemented using *static condensation* for the interior nodes of each element. This reduces considerably the number of global unknowns at an element-by-element cost of generating and solving the small Shur complement system of equations.

The compact discontinuous Galerkin formulation

The CDG formulation requires first to rewrite (2.13a) as a system of first order equations taking into account the discontinuities of the approximation spaces between elements. Thus Equations (2.13) become:

$$\boldsymbol{q} - \beta \mathbf{P} \boldsymbol{\nabla} u = \mathbf{0} \qquad \text{in } \widehat{\Omega}, \tag{2.15a}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{q} + \gamma u = s_x s_y f \qquad \text{in } \widehat{\Omega}, \tag{2.15b}$$

$$[\![\boldsymbol{q} \cdot \boldsymbol{n}]\!] = 0 \qquad \text{on } \Gamma, \tag{2.15c}$$

$$[\![u\, \boldsymbol{n}]\!] = \mathbf{0} \qquad \text{on } \Gamma, \tag{2.15d}$$

$$\boldsymbol{n} \cdot \beta \boldsymbol{\nabla} u - Ik\alpha\beta\, u = g \qquad \text{on } \Gamma_{\mathtt{R}}, \tag{2.15e}$$

$$\boldsymbol{n} \cdot \beta \mathbf{P} \boldsymbol{\nabla} u - Ik\beta\, u = 0 \qquad \text{on } \Gamma_{\mathtt{PML}}. \tag{2.15f}$$

As usual in CDG, proposed by Peraire and Persson (2008), two local lifting operators are defined on all the interior faces (recall that here no Dirichlet boundary conditions are applied). For any interior face $\Gamma_i \subset \Gamma$, the lifting $\boldsymbol{r}^i : [\mathcal{L}^2(\Gamma_i)]^2 \to [\widehat{\mathcal{V}}_h]^2$ is defined by

$$\big(\boldsymbol{r}^i(\boldsymbol{q}), \boldsymbol{v}\big)_\Omega = \big\langle \boldsymbol{q}, \{\boldsymbol{v}\} \big\rangle_{\Gamma_i} \quad \forall \boldsymbol{v} \in [\widehat{\mathcal{V}}_h]^2,$$

Likewise, the second lifting, $\boldsymbol{s}^i : \mathcal{L}^2(\Gamma_i) \to [\widehat{\mathcal{V}}_h]^2$, is defined by

$$\big(\boldsymbol{s}^i(v), \boldsymbol{v}\big)_\Omega = \big\langle v, [\![\boldsymbol{n} \cdot \boldsymbol{v}]\!] \big\rangle_{\Gamma_i} \quad \forall \boldsymbol{v} \in [\widehat{\mathcal{V}}_h]^2,$$

for all interior faces $\Gamma_i \subset \Gamma$.

Following the rationale detailed by Peraire and Persson (2008), the CDG weak form associated to (2.15) requires to find $u_h \in \widehat{\mathcal{V}}_h$ such that

$$\hat{a}(u_h, v) = \hat{l}(v) \quad \forall v \in \widehat{\mathcal{V}}_h,$$

where the bilinear and linear forms, $\hat{a}(\cdot,\cdot)$ and $\hat{l}(\cdot)$ are given by

$$\begin{aligned}\hat{a}(u,v) =& \big(\beta\mathbf{P}\boldsymbol{\nabla}u, \boldsymbol{\nabla}v\big)_\Omega - \big(\gamma u, v\big)_\Omega - \big\langle \beta\mathbf{P}[\![u\boldsymbol{n}]\!], \{\boldsymbol{\nabla}v\}\big\rangle_\Gamma - \big\langle \beta\mathbf{P}\{\boldsymbol{\nabla}u\}, [\![v\boldsymbol{n}]\!]\big\rangle_\Gamma \\ & - I\big\langle k\alpha\beta u, v\big\rangle_{\Gamma_{\mathtt{R}}} - I\big\langle k\beta u, v\big\rangle_{\Gamma_{\mathtt{PML}}} + \big\langle C_{11}[\![u\boldsymbol{n}]\!], [\![v\boldsymbol{n}]\!]\big\rangle_\Gamma \\ & - \big\langle \beta(\mathbf{P}\boldsymbol{C}_{12})\cdot[\![u\boldsymbol{n}]\!], [\![\boldsymbol{n}\cdot\boldsymbol{\nabla}v]\!]\big\rangle_\Gamma - \big\langle \beta(\mathbf{P}\boldsymbol{C}_{12})\cdot[\![v\boldsymbol{n}]\!], [\![\boldsymbol{n}\cdot\boldsymbol{\nabla}u]\!]\big\rangle_\Gamma \\ & + \sum_{\Gamma_i\subset\Gamma} \big(\boldsymbol{r}^e(\beta\mathbf{P}[\![u\boldsymbol{n}]\!]) + \boldsymbol{s}^e(\beta(\mathbf{P}\boldsymbol{C}_{12})\cdot[\![u\boldsymbol{n}]\!]), \boldsymbol{r}^e([\![v\boldsymbol{n}]\!]) + \boldsymbol{s}^e(\boldsymbol{C}_{12}\cdot[\![v\boldsymbol{n}]\!])\big)_\Omega, \\ \hat{l}(v) =& \big\langle g, v\big\rangle_{\Gamma_{\mathtt{R}}} - \big(s_x s_y f, v\big)_\Omega.\end{aligned}$$

Note that in the previous expressions differential operators are assumed to act on these functions piecewise (element-by-element) and not in the sense of distributions.

In CDG, the parameter C_{22} appearing in 2.6 is set to zero, see Remark 3. The parameter C_{11} is non-negative of order $\mathcal{O}(h^{-1})$ and, in absence of Dirichlet boundary conditions, may be considered $C_{11} = 0$ on Γ, see Peraire and Persson (2008). The parameter $\boldsymbol{C}_{12} \in \mathbb{R}^2$ is an additional vector which is defined for each interior face of the domain according to

$$\boldsymbol{C}_{12} = \frac{1}{2}(S_{ij}\boldsymbol{n}_i + S_{ji}\boldsymbol{n}_j)$$

where $S_{ji} \in \{0,1\}$ denotes the switch associated to element Ω_i on the face that element Ω_i shares with element Ω_j. There are several possible choices of the switches, always satisfying $S_{ij}+S_{ji} = 1$. Here the so-called *consistent switch* has been used, see Peraire and Persson (2008) and Cockburn et al. (2002) for details.

Remark 4. *Lifting operators in CDG are associated to individual faces, and therefore there are no connectivities between non-neighbor elements. This induces small stencils as in interior penalty methods, see the comparison by Montlaur et al. (2010). However, the connectivity between the interior nodes of one element and interface unknowns of neighboring elements, precludes the possibility of static condensation in CDG.*

The hybridizable discontinuous Galerkin formulation

Following the derivation explained in Section 2.1, the HDG method for system (2.13) or, more precisely, for (2.15) can be stated as: find $(\boldsymbol{q}_h, u_h, \hat{u}_h) \in [\widehat{\mathcal{V}}_h]^2 \times \widehat{\mathcal{V}}_h \times \widehat{\Lambda}_h$ such that

$$\left.\begin{aligned} \big(\mathbf{Q}\boldsymbol{q}_h, \boldsymbol{v}\big)_{\Omega_i} + \big(u_h, \boldsymbol{\nabla}\cdot\boldsymbol{v}\big)_{\Omega_i} - \big\langle \hat{u}_h, \boldsymbol{v}\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_i} &= 0 \\ \big(\boldsymbol{\nabla}\cdot\boldsymbol{q}_h, v\big)_{\Omega_i} + \big(\gamma u_h, v\big)_{\Omega_i} - \big\langle \tau(u_h-\hat{u}_h), v\big\rangle_{\partial\Omega_i} &= \big(s_x s_y f, v\big)_{\Omega_i} \end{aligned}\right\} \text{ for } i = 1,\ldots,\mathtt{n_{el}}, \tag{2.16a}$$

$$\sum_{i=1}^{\mathtt{n_{el}}} \big\langle \big(\boldsymbol{q}_h\cdot\boldsymbol{n} - \tau(u_h-\hat{u}_h)\big), \mu\big\rangle_{\partial\Omega_i} - I\big\langle k\alpha\beta\,\hat{u}_h, \mu\big\rangle_{\Gamma_{\mathtt{R}}} - I\big\langle k\beta\,\hat{u}_h, \mu\big\rangle_{\Gamma_{\mathtt{PML}}} = \big\langle g, \mu\big\rangle_{\Gamma_{\mathtt{R}}}, \tag{2.16b}$$

for all $(\boldsymbol{v}, v, \mu) \in [\widehat{\mathcal{V}}_h]^2 \times \widehat{\mathcal{V}}_h \times \widehat{\Lambda}_h$, where a new matrix is defined $\mathbf{Q} := (\beta\mathbf{P})^{-1}$. As noted in 2.1, the *trace* variable $\hat{u}_h \in \widehat{\Lambda}_h$, which is single valued in each face of the mesh, is an approximation of the trace of the solution u on the mesh faces.

Details of the implementation of HDG for the Mild Slope equation can be found in Appendix B.

The coefficient τ may be prescribed as a positive value on every face of each triangular element (*all faces* approach), see Cockburn et al. (2009) and Kirby et al. (2011), or may be set to zero except on a single arbitrary chosen face of each element (*single face* approach), see Cockburn et al. (2008). Both options require τ to be large enough for stability, and both provide, for properly chosen values of τ, optimal convergence in the solution u and its gradient $\boldsymbol{q}$, viz. order $p+1$ in $\mathcal{L}_2$ norm. More precisely, extensive numerical evidence by Cockburn et al. (2008) and Kirby et al. (2011) shows that values of order one provide optimal behavior for a dimensionless problem. Hence, the same postprocess defined in 2.1 leads to a superconvergent solution also for problem (2.15). Nevertheless, the single face approach presents an increased robustness for the choice of τ, it allows using larger values without any remarkable impact in the solution. This is not the case for the all faces approach. When large values of τ are prescribed in every face superconvergence of the postprocessed solution can be lost.

Note that this behavior induces an extra difficulty for equations with non-constant coefficients, such as (2.11), because τ should vary in each element to account for the variability of the coefficients. In conclusion, the single face stabilization parameter is considered here, that is

$$\tau = \begin{cases} 0 & \text{on } \partial\Omega_i \backslash \partial\Omega_i^\tau \\ \tau_i & \text{on } \partial\Omega_i^\tau \end{cases} \text{ for } i = 1, \ldots, \mathtt{n_{el}}, \tag{2.17}$$

with a constant $\tau_i > 0$ and $\partial\Omega_i^\tau$ an arbitrary face of element Ω_i. A constant value τ_i is used for all the elements, and a simple dimensional analysis leads to the following expression for the minimum value of τ_i to be used

$$\tau_i \geq \tau^- := \max_{\Omega} \{k\beta |s_x/s_y|\} \text{ for } i = 1, \ldots, \mathtt{n_{el}}, \tag{2.18}$$

which would correspond to $\tau^- \approx 1$ for a dimensionless problem with constant coefficients, see Appendix A.. Note that τ is defined element-by-element, thus a face shared by two elements may have two different values of τ, see also Remark 3.

From a computational point of view, there are $\mathtt{n_{el}}$ local problems corresponding to (2.16a), which are equivalent to solving the elemental Schur complements (static condensation) for the interior nodes in CG. The global solve in HDG, which corresponds to compute the trace $\hat{u}_h$ from (2.16b), has more unknowns than CG; note that $\widehat{\Lambda}_h$ is a larger space than the corresponding CG traces space, because all the vertices have a multiplicity equal to the number of faces connected to them, see Figure 2.3. However, HDG has a uniform block structure, which can be exploited by the solver to improve its efficiency. Moreover, the weight of the multiplicity of the vertices decreases as the approximation p is increased. In any case, it is also obvious from Figure 2.3 that HDG has less DOF than CDG.

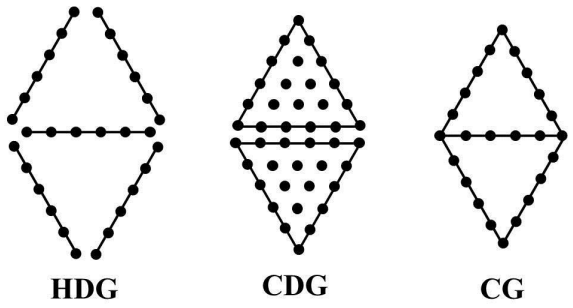


Figure 2.3: Degrees of freedom in a two triangular elements mesh of degree 5 for HDG, CDG and CG.

2.2.3 Numerical results

Three examples are used to compare the performance of low and high-order CG, CDG and HDG. The first one has an analytical solution allowing for a thorough error analysis. The others, which include a real engineering application, use a reference overkilled solution (high-order CG with a refined mesh) to evaluate the precision of the different approximations. Note that in all these comparisons the PML region is kept unmodified to minimize its influence on the analysis. Moreover, when comparing HDG the superconvergent solution described in Section 2.1 is used.

To compare the performance of the different Galerkin methods two basic criteria are employed: the number of DOF (more precisely, the size of the global system to be solved) and the total runtime. These comparison are done taking into account the precision of the results, the element characteristic size h, and the degree of the approximation p, which can be combined in the dimensionless wavenumber (i.e. number of nodes per wavelength), see Remark 5.

Obviously, the computational time strongly depends, among others, on the quadrature rules employed. Here, Wandzurat symmetric rules for triangles are considered, see Wandzurat and Xiao (2003). Moreover, in order not to penalize low-order elements, some Dunavant rules for low orders are also implemented, see Dunavant (1985). Table 2.1 specifies the number of integration points corresponding to each order of the quadrature employed in the following examples. In practice, given the

Table 2.1: Order of the quadrature rules employed with their corresponding number of integration points.

Order of quadrature rule	2	3	5	6	10	15	20	25	30
Number of integration points	3	4	7	12	25	85	85	126	175

required integration order defined by the corresponding weak form and the degree of approximation p, the rule adopted is the lowest possible of those listed in Table 2.1. For orders of quadratures larger than 30 no Wandzurat quadrature is available, and a tensor quadrature is used in this case.

Remark 5 (*dimensionless wavenumber*)**.** *The so-called dimensionless wavenumber $\xi := kh/p$ will be used to quantify the wave resolution. For example, $\xi = \pi/4$ corresponds to 8 linear elements per wavelength and it is considered a minimum resolution for linear elements, see Donea and Huerta (2003). However, as already noted by Thompson and Pinsky (1994) high-order elements require less nodes per wavelength.*

Homogeneous circular scattering of a plane wave

A standard benchmark for wave problems is considered, see for example Bowman et al. (1987): a plane wave is scattered by a cylindrical object of unitary radius in a homogeneous media. In this case, the problem defined in (2.13) is simplified because β is constant and $f = 0$. Given the symmetry of the problem, only half of the geometry is simulated as depicted in Figure 2.4. An analytical solution to this problem is given by MacCamy and Fuchs (1954).

A study of the accuracy of the approximate solution is performed using unstructured triangular meshes for CG, CDG and HDG, and three different values of the wavenumber, $k = 11$, 25, and 100, corresponding to a number of waves in the domain equal to 17, 40 and 160 respectively. Thus, the range considered is in the mid-high frequency regime. As an example of the obtained results, Figure 2.4 depicts the solution for $k = 25$. Four different values of the polynomial degree p are tested, $p = 1$, 2, 5, and 9. For each combination of k and p, four computations are carried out for each method, with mesh sizes corresponding to the dimensionless wavenumbers $\xi = 0.5$,

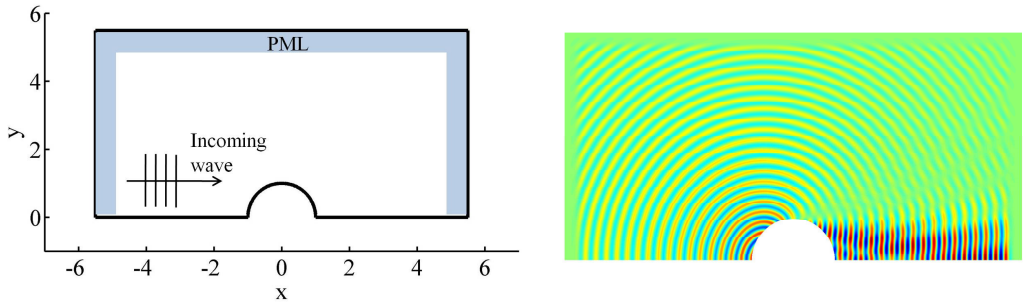


Figure 2.4: Circular scattering of a plane: problem statement (left) and scattered wave for $k = 25$ (right).

0.75, 1, and 1.25, see Remark 5. The corresponding wave resolutions are 12.6, 8.4, 6.3, and 5.0 nodes per wavelength. Thus, a total of 144 simulations have been carried out. Note that for a given ξ and wavenumber k, the meshes corresponding to different p have approximately the same number of nodes in the domain.

The quantity kR, where R is the radius of the scattering cylinder, represents in this problem a characteristic number, being related to the ratio between the dimension of the scatterer and the wavelength. Figure 2.5 depicts the relative $\mathcal{L}^2$ error as a function of the element size, for CG, CDG and HDG, with $kR = 11$ and $kR = 100$. The relative $\mathcal{L}^2$ error is evaluated over the computational domain (top) —not including the PML region— and on the scattering boundary (bottom). The mean slope of the curve is displayed at the bottom of each curve. For high-order approximations, it is clear that the slopes of the convergence curves increase with the wavenumber indicating that the dispersion error is dominating, see (2.10). On the contrary, low order approximations seem less sensible to dispersion, which is conforming with the results by Feng and Wu (2009) and Alvarez et al. (2006) for the interior penalty discontinuous Galerkin.

It is important to note that Figure 2.5 indicates that HDG always induces smaller errors than CG in the interior and on the scattering boundary on a given discretization (same h and p). This is not the case for CDG at $p = 9$.

The number of DOF, i.e. the global linear system dimension, for a fixed value of the error is depicted in Figure 2.6. The mesh size h for each accuracy level is estimated from the convergence curves in Figure 2.5 and additional computations

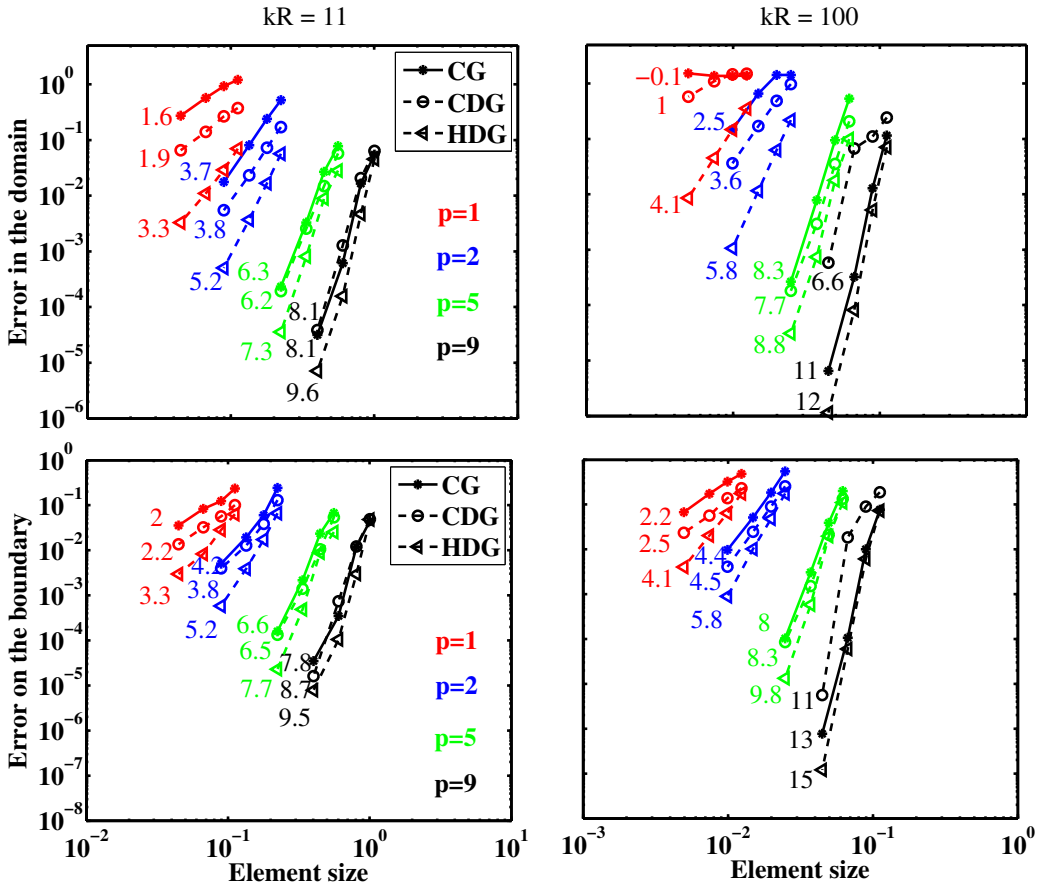


Figure 2.5: Circular scattering of a plane wave: convergence versus element size for two different values of wave length $kR = 11$ (left) and $kR = 100$ (right) and over the computational domain (top) or the scattering boundary (bottom).

with $k = 25$. More specifically, the mesh size is linearly interpolated in log-log scale, when the chosen error level is in between the limits of the convergence curve, or it is linearly extrapolated using the closest two points.

The curves in Figure 2.6 illustrate how an increase in p reduces the number of nodes per wavelength for a given precision for every method. This is in perfect agreement with Thompson and Pinsky (1994) and Ihlenburg and Babuška (1997). Note that for a given frequency the reduction in number of DOF is of several orders of magnitude for mid to high accuracies. Similar conclusions are inferred analyzing the error in the domain or the error on the scattering boundary.

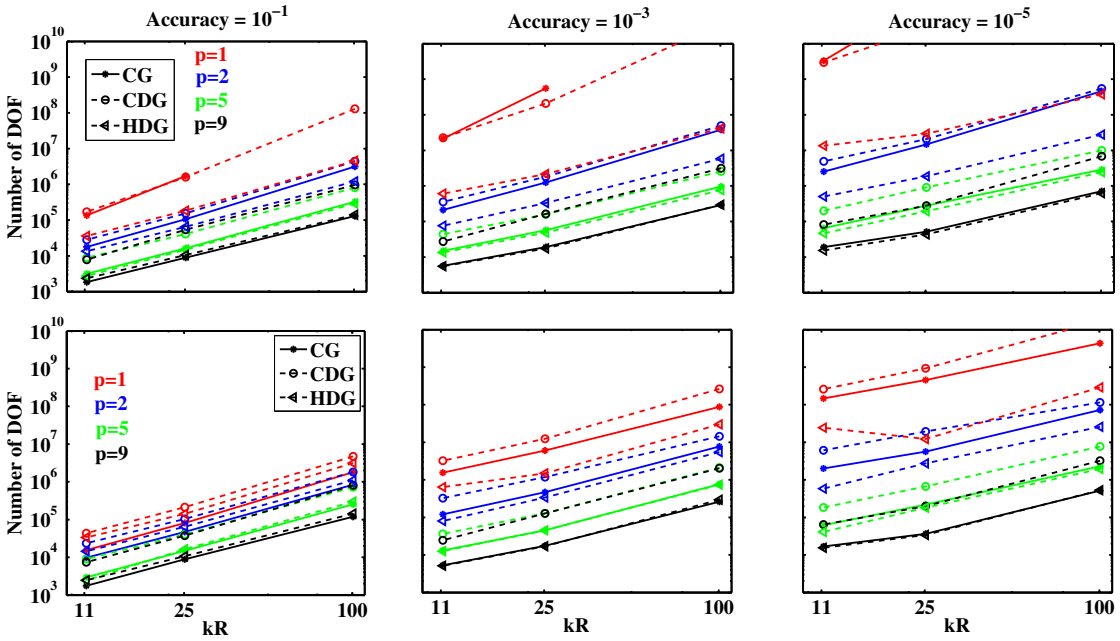


Figure 2.6: Circular scattering of a plane wave: DOF for an accuracy of 1e-1 (left), 1e-3 (center), 1e-5 (right), for the error in the domain (top) and on the scattering boundary (bottom).

For CG and HDG, the reduction in number of DOF for increasing p is more important because of their lack of interior nodes. In fact, due to the similarity between static condensation and the hybridization technique, CG and HDG perform similarly, with almost the same number of DOF for a given accuracy level for high-order approximations. For low-order approximations $p = 1, 2$, HDG method performs better than CG *without stabilization*, due to the fact that DG methods are less sensitive to dispersion errors.

As expected, Figure 2.6 also shows that CDG requires considerable more DOF than HDG or CG for the same level of accuracy, due to the coupling of the nodes in the interior of the element with neighboring element nodes. Thus, the computational cost for the linear system solution is clearly larger for CDG than for CG or HDG. In fact, the computational cost for the linear system solution is related not only to the dimension of the linear system but also with the number of non-zero entries of the linear system matrix. These two factors penalize CDG.

To conclude, given that CG and HDG exhibit similar efficiency clearly outperforming CDG, CDG is discarded and will not be considered in the following tests, focusing the discussion in CG and HDG.

Next, CG and HDG are further compared in terms of runtime. The total time to obtain the solution is considered: matrix generation and assembly, linear system solution, and the evaluation of the solution at the interior nodes in CG and HDG. In this latter case the postprocess for obtaining the superconvergent approximation, see Section 2.1, is also taken into account.

As noted earlier the assembly time is largely influenced by the quadrature rules implemented. The order of the quadrature rules considered are those needed to exactly integrate a *mass* matrix of a straight-sided element, which corresponds to the second term in (2.13a) and, for this example, with constant coefficients. The overhead due to the integration on curved elements and PML elements (for which non-constant coefficients are present) is neglected, given that curved elements and PML elements are a small percent of the whole element set. For a given p, the quadrature rule adopted is the one with the lowest order that is $\geq 2p$ in Table 2.1.

All computations are carried out with Matlab running on a Xeon E5640, 2.66 Ghz/12MB cache with 72 GB of RAM. A code optimization based on the substitution of the code loops by matrix-matrix multiplication (whenever possible) has been adopted, see Roca et al. (2011) for a detailed description of this optimization. For the linear system solution, a direct solver is considered. In contrast with the comparison by Kirby et al. (2011) done with an hermitian positive definite matrix, here the complex matrix is symmetric but not hermitian because of the boundary term in (2.14) and (2.16) and the complex coefficients in the PML region. Thus, a general LU factorization with partial pivoting is used.

In Figure 2.7 the relative $\mathcal{L}^2$ error in the domain is depicted as a function of the total computing time for CG and HDG, for $kR = 11$ and $kR = 100$. These results suggest that for a given accuracy CG is faster than HDG, at least for high-order approximations. To further analize this issue and to confirm the better performance of high-order approximations, these curves, and also additional results with wavenumber

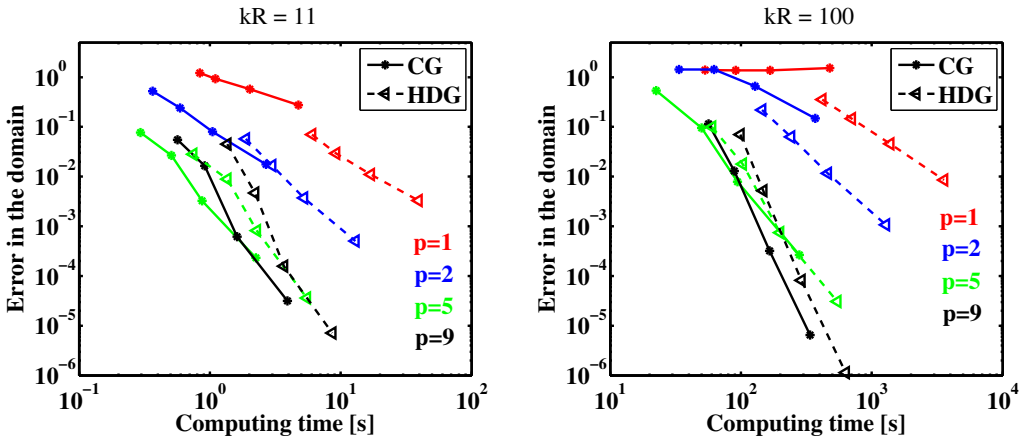


Figure 2.7: Circular scattering of a plane wave: error in the domain vs runtime for CG and HDG and $kR = 11$ (left) and $kR = 100$ (right).

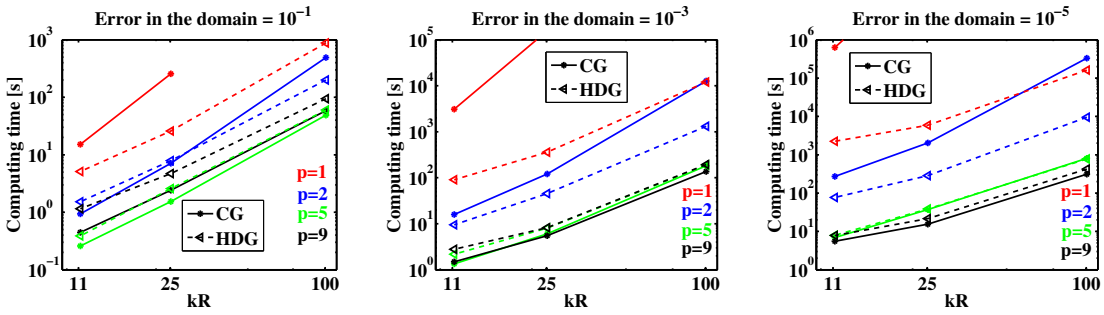


Figure 2.8: Circular scattering of a plane wave: runtime versus wavenumber at different precisions 1e-1 (left), 1e-3 (center), and 1e-5 (right).

$k = 25$, are interpolated to obtain values of the computational time for fixed values of the error, as shown in Figure 2.8.

It is important to note that for a given accuracy, high-order elements require less computing time than low-order ones, also at low accuracy. Note also that high-order computations $p = 5$ and $p = 9$ induce, in general, a similar performance. For low precision $p = 5$ outperforms $p = 9$ but as the accuracy is tightened $p = 9$ is faster than $p = 5$. As expected, the best order p depends on the actual problem to be solved. But, as a general rule, high-order elements ($p > 2$) are more efficient than low order ones, clearly outperforming the usual practice of linear approximations.

With respect to CG and HDG several observation can be extracted. Obviously,

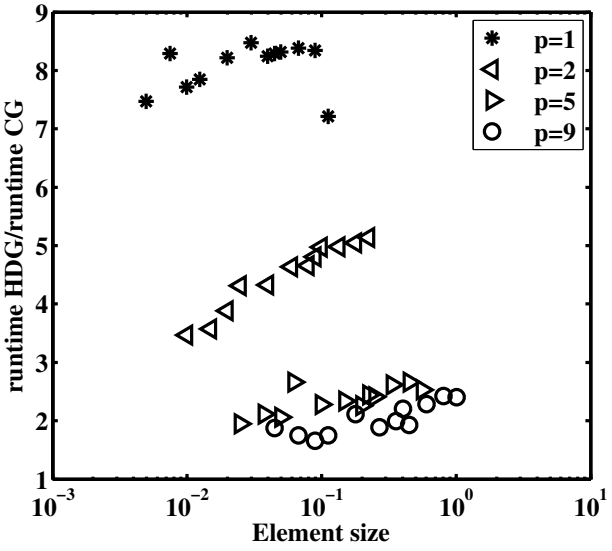


Figure 2.9: Circular scattering of a plane wave: runtime ratio between HDG and CG for fixed meshes.

for $p = 1$, HDG outperforms CG because no stabilization is used in CG whereas HDG incorporates it intrinsically. For $p = 2$, except for low accuracies and large wavelengths where both methods are almost equivalent, HDG is faster than CG for given precision and this improvement of runtime in HDG is more important as k increases. This is not the case for high-order approximations. For $p = 5$ and 9 at a given precision, CG is faster than HDG. This is more obvious for low accuracies. As the precision increases both performances are similar.

In fact, it is worth noting that the overhead of HDG for a given discretization (given p and h) has a simple behavior. Figure 2.9 clearly shows that the runtime overhead for a fixed mesh is almost independent of the characteristic element size and that it decreases as the degree of the approximation increases. For high degrees this overhead is almost constant for p. Consequently, for a given high-order CG discretization with a uniform characteristic mesh size, from Figure 2.9 and the Remark 6, one can compute the characteristic size of HDG inducing a similar computational cost.

Note however, as observed earlier, that the HDG overhead can be compensated at mid to high accuracies by its increased precision. In fact, recall that as the precision

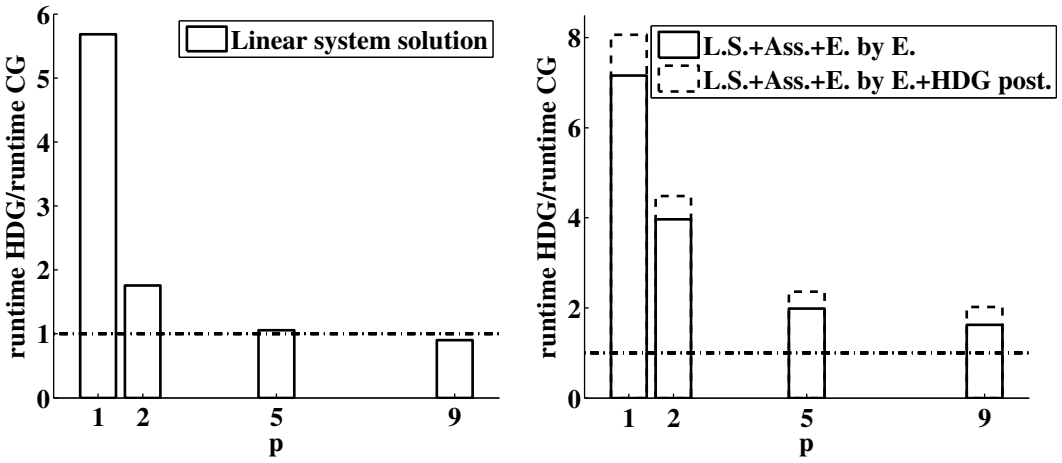


Figure 2.10: Circular scattering of a plane wave: runtime ratio between HDG and CG for fixed meshes only for the linear solve (left) and accounting for all element computations (right).

increases CG and HDG show similar performances.

Remark 6 (Runtime)**.** *Numerical evidence, as shown in this example, indicates that for CG and HDG and meshes having a number of boundary faces negligible compared with the number of interior ones, i.e. boundary influence is negligible, the runtime is approximately* $\mathcal{O}(n_{el}) \approx \mathcal{O}(h^{-2})$.

To further analyze the overhead of HDG respect to CG for a fixed mesh, Figure 2.10 shows the runtime ratio between HDG and CG. The left figure shows the ratio only for the linear system solve and the right one the total runtime ratio, i.e. linear solve plus all element computations: creation and assembly of matrices, element-by-element solution —Schur complement for CG and local solve for HDG, i.e. Equations. (2.16a)— and postprocess for HDG. Given the relative independence of the runtime overheads on the mesh size shown in Figure 2.9, Figure 2.10 shows averages at each degree p for every computed mesh size h.

The remarkable overhead of HDG for linear elements compared to CG is drastically reduced as p increases. In fact, in agreement with Kirby et al. (2011), HDG requires less runtime to solve the global linear system for high-orders. This can be explained by the uniform block-structure of HDG, which in average induces a smaller bandwidth

compared to CG, see Kirby et al. (2011) for details.

The local (elemental) computations, on the other hand, are always more costly for HDG, due its increased complexity in the local problems. This is shown in Figure 2.10 (right). In spite of its faster linear solve, HDG is still more expensive than CG even for $p = 9$ when the total cost is considered, that is, adding to the global linear solve the time required for the assembly and the local element-by-element operations. Note however, all elemental computations are likely to be parallelized, reducing the weight of the elemental computations on the global solving time.

Finally, Figure 2.10 (right) also depicts the overhead associated to the postprocess of the HDG solution which allows to compute a superconvergent approximation. This is to show that the extra cost is minor compared to the advantage of obtaining a solution with a higher precision or, simply, an estimation of the error.

Rectangular cavity problem

In this dimensionless example an incoming plane wave at angle 225° with respect to the x-axis and a wavelength of 0.25 is propagated in a semi-plane with a squared notch. As in the previous example, the scattered wave is the solution of Equation (2.13) with constant coefficient β and $f = 0$. The value of the absorption parameter used to model the boundary is $\alpha = 0$ inside the notch (i.e. totally reflecting) and $\alpha = 1$ (absorbing) on the rest of the physical boundary. Figure 2.11 shows the problem statement and the scattered wave. The geometry is similar to the one used by Panchang et al. (1991) and ideally models the resonance of sea waves in a rectangular harbor with constant bathymetry. The rectangular closed region is considered as the zone of interest of the domain, where wall reflections generate a complex wave interaction pattern that makes the test particularly challenging from a computational point of view. In addition, as usually in the presence of non-smooth boundaries such as corners, the solution of the wave equation is singular leading to non-optimal convergence both for h and p-refinement, see for instance Babuška and Szabo (1991) and Costabel and Dauge (2000). This test aims to show that, in the absence of adaptive mesh refinement, high-order elements are more efficient than low-order ones, even in the presence of

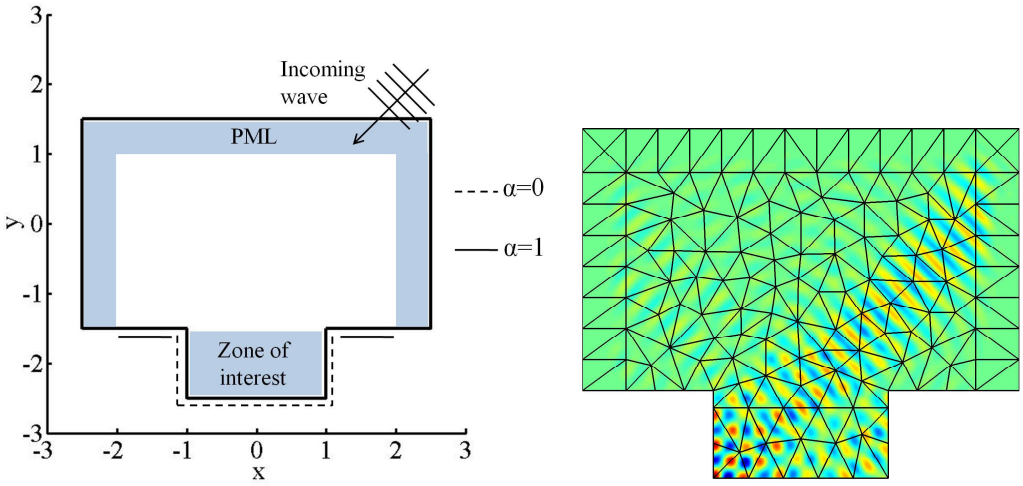


Figure 2.11: Rectangular cavity: problem statement (left) and scattered wave (right) on a mesh with $h = 0.4$, $p = 8$, and $\xi = 1.26$.

singularities.

Four different polynomial degree are tested, $p = 1$, 2, 4, and 8. For each degree four nested meshes are used, they correspond to $\xi = 1.257$, 0.628, 0.314, and 0.157 (i.e. 5, 10, 20, and 40 nodes per wavelength). Given that no analytical solution is available for this test, the relative $\mathcal{L}^2$ error in the zone of interest is evaluated comparing the solution with a reference solution computed with CG with an h-refined high-order mesh, with 160 nodes per wavelength.

Figure 2.12 depicts for CG and HDG the relative $\mathcal{L}^2$ error as a function of the number of nodes per wavelength, i.e. the dimensionless wavenumber ξ (left), and as a function of the computing time (right). In the presence of singularities, as expected, all curves asymptotically reach the same slope, see Babuška and Szabo (1991), as the dimensionless wavenumber decreases (as the number of nodes per wavelength increases). In any case, high-order elements require less nodes per wavelength than low-order elements for a given accuracy. Thus, high-order approximations require less DOF for a given accuracy. This has a direct impact in the computing time as shown also in Figure 2.12 and, for a given accuracy, high-order elements are faster than low-order ones. Thus, it is worth noting that, in spite of the singularities, high-order approximations outperform low-order ones. Obviously, an *hp* strategy would improve

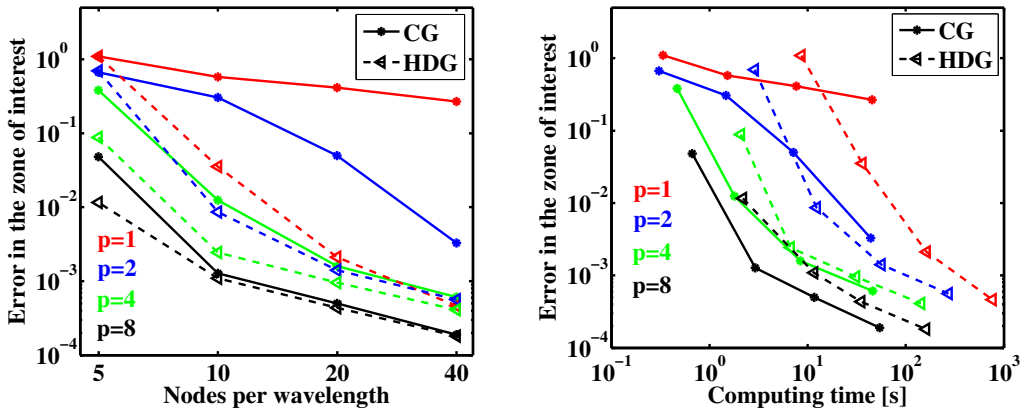


Figure 2.12: Rectangular cavity: error versus nodes per wavelength (left) and runtime (rigth).

the performances, but this is out of the scope of the present comparison.

The performance of CG for $p = 1$ is clearly hampered by the lack of stabilization. The comparison between CG and HDG should be focused on the other orders. For $p = 2$ except for very low accuracies (below 0.1) HDG outperforms CG in runtime. Nonetheless, CG proves to be faster than HDG for a given accuracy for high-order elements. Obviously, the overhead of HDG decreases as the accuracy increases.

Barcelona harbor

In this section, the Mild Slope equation is used to study the wave propagation in Barcelona's harbor. The Mild Slope equation, derived by Berkhoff (1972), allows to simulate the wave propagation from the open sea until the interior of the harbor, and it is considered accurate for bottom slopes less or equal to 1/3, see Booij (1983). Details on the derivation can be found in Appendix F.

Thus, in this example, the coefficient β in problem (2.13) is non-constant and depends on the bathymetry, see Berkhoff (1972) for details. Moreover, the incident potential,

$$u_0 = \exp\big(ik_0(x\cos\theta_0 + y\sin\theta_0)\big),$$

with $k_0 = 0.1121\text{m}^{-1}$ (corresponding to a wavelength of 56m) and angle of incidence

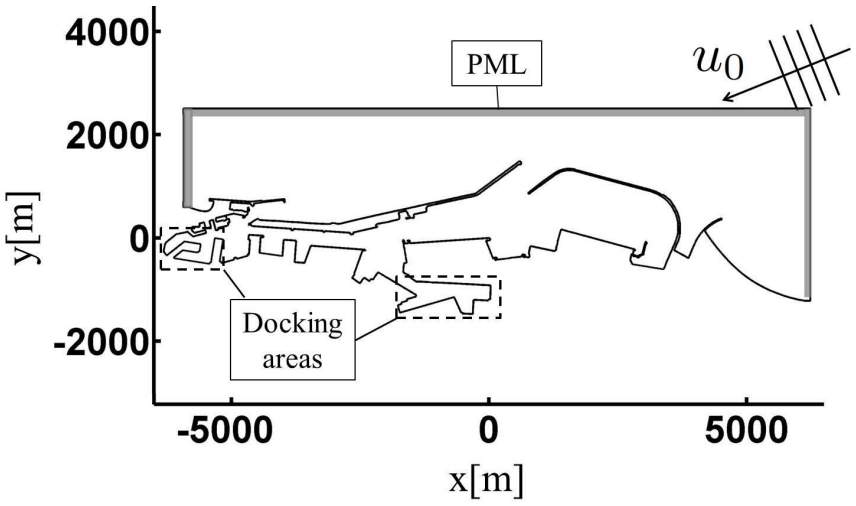


Figure 2.13: Barcelona harbor: statement of the problem.

$\theta_0 = 202.4°$ from the x-axis, induces a non-homogeneous equation, namely

$$f = -\boldsymbol{\nabla} \cdot (\beta \nabla u_0) - k^2 \beta u_0.$$

Figure 2.13 shows the computational domain, it ranges in latitude between 41.28° N and 41.38° N and covers an area of about 34.4km^2. The PML area is also depicted as well as the areas of interest. The bathymetry data comes from a real measure campaign[1], and it has been modified to comply the condition of constant bottom depth in the PML area. This condition is sufficient to ensure absorption of the PML media, see for example Kucukcoban and Kallivokas (2011). The absorption coefficient α on physical boundaries has been set to 0.05 for dikes, 0.43 for breakwaters and 0.74 for beaches.

The usual output of interest for this problem is the so-called *wave amplification factor*, that is, the ratio between the total and incident wave-height, namely

$$H = \frac{|u + u_0|}{|u_0|}. \tag{2.19}$$

[1]Data provided by the *Autoritat Portuaria de Barcelona*

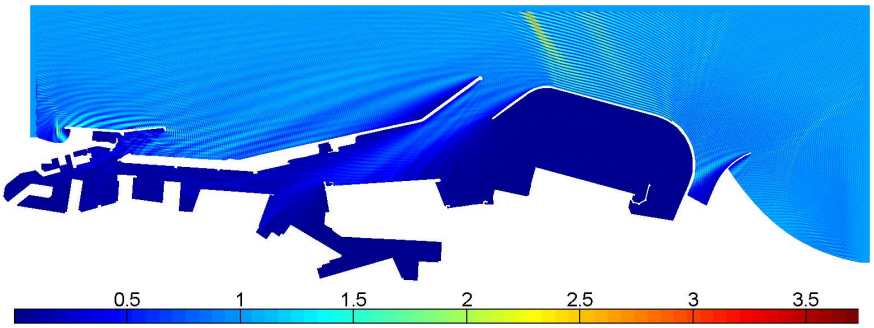


Figure 2.14: Barcelona harbor: wave amplification factor.

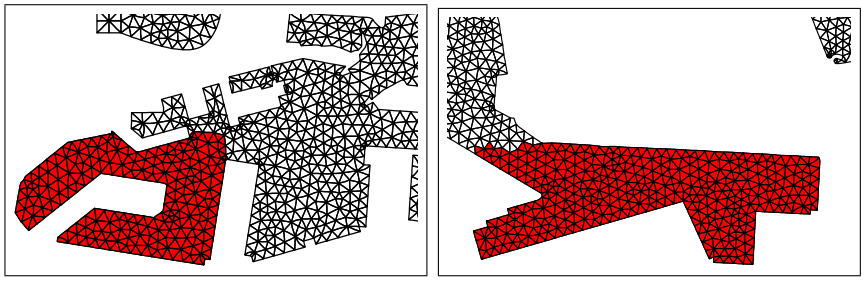


Figure 2.15: Barcelona harbor: zoom in the docking areas.

It is plotted for this case in Figure 2.14. Note that, with the selected period of the incident wave, more than 200 waves are present in the computational domain.

The areas of interest, denoted as $\Omega_{\texttt{int}}$, already shown in Figure 2.13 are zoomed in Figure 2.15. The error in $\Omega_{\texttt{int}}$ is computed using a reference solution. Two measures of the error are considered, that is, the mean error

$$E^2 = \frac{1}{\text{meas}(\Omega_{\texttt{int}})} \int_{\Omega_{\texttt{int}}} (H^* - H)^2 \, d\Omega,$$

and the maximum elemental error, that is

$$E^2 = \max_{\forall \Omega_i \in \Omega_{\texttt{int}}} \frac{1}{\text{meas}(\Omega_i)} \int_{\Omega_i} (H^* - H)^2 \, d\Omega,$$

where H^* is the wave amplification factor computed with the reference solution.

Tables 2.2 and 2.3 summarize different indicators (mean error in the area of interest, maximum elemental error in the area of interest, dimension of the global system,

Table 2.2: Barcelona harbor: error, DOF and runtime for CG.

h [m]	4	10	10	50	50	50
p	2	2	4	8	10	12
error	0.28e-2	0.26e-1	0.28e-3	0.14e-1	0.21e-2	0.50e-3
max el.err.	0.16e-1	0.14	0.15e-2	0.38e-1	0.87e-2	0.22e-2
DOF	4 908 818	1 647 698	4 115 482	367 198	467 198	567 198
runtime [s]	318	80	515	79	168	485

Table 2.3: Barcelona harbor: error, DOF and runtime for HDG.

h [m]	4	10	10	10	50	50	50
p	2	1	2	4	8	10	12
error	0.78e-4	0.33e-1	0.79e-3	0.23e-3	0.30e-2	0.57e-3	0.29e-3
max el.err.	0.98e-3	0.13	0.35e-2	0.95e-3	0.95e-2	0.30e-2	0.16e-2
DOF	11 033 058	2 467 784	3 701 676	6 169 460	$4.5\,10^5$	$5.5\,10^5$	$6.5\,10^5$
runtime [s]	2 426	435	671	1 722	224	486	1125

and runtime) for CG and HDG and different discretizations. Three computational meshes with characteristic element size $h = 4$m, 10m, and 50m have been considered, with 2 446 554, 820 086, and 32 802 elements, respectively. Obviously, smaller elements are present in the interior of the harbor to properly capture geometric details. As an example, Figure 2.15 also shows in the area of interest the mesh of characteristic size $h = 50$m. For this mesh, the smoothing technique proposed by Roca et al. (2011) has been used for untangle elements and improve the quality of the mesh.

It is worth noting that the wavenumber depends on the bottom depth through the so-called dispersion relation, leading to a non-uniform dimensionless wavenumber ξ, see Remark 5, for a fixed element size h. As an example, the dimensionless wavenumber for the discretization with characteristic element size $h = 10$m could vary for a degree $p = 10$ from $\xi = 0.2$ (30 nodes per wavelength) in the docking areas with bottom depth of about 3m, to $\xi = 0.11$ (57 nodes per wavelength) in open sea with bottom depth of 30m.

As noted earlier, if runtime must be estimated the quadrature rule must determined. In this example, the coefficients are non-constant. Using quadratures (with the minimum number of integration points) from Table 2.1 able to integrate exactly

polynomials up to degree $3p$, gives sufficient resolution and allows a fair comparison of runtime between CG and HDG.

Results in Tables 2.2 and 2.3, corroborate that both CG and HDG perform better with large high-order elements than smaller low-order ones. As in the previous examples, the use of large high-order elements reduces the computational cost for a given level of accuracy.

For instance, comparing CG with $\{h = 4, p = 2\}$ and $\{h = 50, p = 10\}$ it can be noticed that both have a similar mean error in the area of interest but $\{h = 4, p = 2\}$ has a larger maximum error, ten times more DOF and twice runtime. High-order methods also perform better for lower accuracies, one order lower when $\{h = 10, p = 2\}$ is compared with $\{h = 50, p = 8\}$ in the same table. In this case the runtime is similar but the high-order approximation induces smaller errors, in particular, the maximum one.

Table 2.3 corroborates the same conclusions for HDG. For instance, HDG solutions with $\{h = 10, p = 2\}$ and $\{h = 50, p = 10\}$ have similar accuracies (in terms of mean or maximum error) but the high-order approach is 1.4 times faster. When comparing $\{h = 10, p = 4\}$ and $\{h = 50, p = 12\}$ the ratio is also similar, 1.5 time faster the high-order approximation.

Note also that, as expected, the comparison of Tables 2.2 and 2.3 shows once again how HDG provides better accuracy than CG for the same computational mesh, but with an overhead in runtime, which decreases as the degree of approximation increases (recall Figure 2.10). This motivates to compute the characteristic size of HDG inducing a similar computational cost for a given high-order CG discretization, as stated in Section 2.2.3. Table 2.4 shows precisely the results of CG for $h = 50$ and those of HDG on a mesh with a characteristic size $h = 84$, which is the value obtained from Figure 2.9 and Remark 6. Consequently, the computational costs displayed in Table 2.4 are similar for CG and HDG. This table confirms that, for a similar computational cost, CG and HDG have similar accuracy. Nevertheless, p-adaptive strategies are more easily implemented in an HDG framework, see Giorgiani et al. (2012), than in a CG code.

Table 2.4: Barcelona harbor: analogous CG and HDG results.

	p	8	10	12
CG with $h = 50$m	error	0.14e-1	0.21e-2	0.50e-3
	max el. error	0.38e-1	0.87e-2	0.22e-2
	DOF	367 198	467 198	567 198
	runtime [s]	79	168	485
HDG with $h = 84$m	error	0.13e-1	0.11e-2	0.70e-3
	max el. error	0.60e-1	0.44e-2	0.25e-2
	DOF	164 259	200 761	237 263
	runtime [s]	82	175	408

2.3 High-order HDG for transient Navier-Stokes

In this section, the HDG method is specialized to the solution of incompressible flows. In Section 2.3.1 the discrete equations are presented and a 2D steady state application is shown in Section 2.3.2. The goal is to show that high-order elements are convenient respect to low-order elements also for fluid problems, in particular for incompressible flows where no shocks appears in the solution, hence smooth fields can be considered.

2.3.1 The HDG system for the Navier-Stokes equations

Let $\Omega \in \mathbb{R}^d$ be an open bounded domain with boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$. The strong form of the time dependent Navier-Stokes equations with Dirichlet and Neumann boundary conditions, and symmetric form of the convective term, is

$$\begin{aligned}
\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{u}) + \boldsymbol{\nabla} p - \nu\Delta\boldsymbol{u} &= \boldsymbol{b} && \text{in } \Omega,\ t>0,\\
\boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0 && \text{in } \Omega,\ t>0,\\
\boldsymbol{u} &= \boldsymbol{g} && \text{on } \partial\Omega_D,\ t>0,\\
(-p\mathbf{I} + \nu\boldsymbol{\nabla}\boldsymbol{u})\boldsymbol{n} &= \mathbf{0} && \text{on } \partial\Omega_N,\ t>0,\\
\boldsymbol{u} &= \boldsymbol{u}_0 && \text{in } \Omega,\ t=0,
\end{aligned} \tag{2.20}$$

where $\boldsymbol{u}$ and p are the velocity and the pressure in the fluid, ν is the kinematic viscosity, $\boldsymbol{b}$ is a body force and $\boldsymbol{n}$ is the unitary outward normal vector. Proper initial

conditions $\boldsymbol{u}_0$ are assumed, and $\boldsymbol{g}$ is the prescribed velocity on the Dirichlet boundary $\partial\Omega_D$. Pseudo-stress free boundary conditions are imposed in the Neumann boundary $\partial\Omega_N$, see Donea and Huerta (2003), however more *physical* conditions can also be used. Nguyen et al. (2011c) proposed a survey on the implementation of different kind of boundary conditions for the Navier-Stokes equations in HDG.

System 2.20 is rewritten as a system of first order equations, introducing the new variable $\boldsymbol{L}$ corresponding to the velocity gradient. The discontinuities of the approximation spaces between elements is also taken into account. Thus Equations (2.20) become

$$\boldsymbol{L} - \boldsymbol{\nabla}\boldsymbol{u} = \boldsymbol{0} \qquad \text{in } \widehat{\Omega}, \tag{2.21a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{u} + p\mathbf{I} - \nu\boldsymbol{L}) = \boldsymbol{b} \qquad \text{in } \widehat{\Omega}, \tag{2.21b}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 \qquad \text{in } \widehat{\Omega}, \tag{2.21c}$$

$$[\![\boldsymbol{u}\otimes\boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma, \tag{2.21d}$$

$$[\![(-p\mathbf{I} + \nu\boldsymbol{L})\boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma, \tag{2.21e}$$

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \partial\Omega_D,$$

$$(-p\mathbf{I} + \nu\boldsymbol{L})\boldsymbol{n} = \boldsymbol{0} \qquad \text{on } \partial\Omega_N,$$

where $\mathbf{I}$ is the identity matrix of dimension $d \times d$. Equations (2.21d) and (2.21e) impose the continuity of the velocity and the normal pseudo-stress across interior element faces. Thus, the incompressible Navier-Stokes equations are stated element-by-element in the broken domain $\widehat{\Omega}$, see Equations (2.21a), (2.21b) and (2.21c).

The main ingredient of the HDG method is the introduction of an approximation of the trace of the velocity on the mesh skeleton Γ, $\hat{\boldsymbol{u}}$. This new variable allows to state a local problem in each element Ω_i, corresponding to the Navier-Stokes equations (2.21) with Dirichlet boundary conditions

$$\boldsymbol{u} = \hat{\boldsymbol{u}} \ \text{ on } \partial\Omega_i. \tag{2.22}$$

The Dirichlet local problem in each element can be closed prescribing, for instance, the pressure at one point or the mean value for the pressure in the element. The usual choice in HDG is prescribing the mean pressure in the element boundary, that is

$$\big\langle p_h, 1\big\rangle_{\partial\Omega_i} = \rho_{h,i}, \quad i = 1, \ldots, \mathtt{n_{el}}.$$

Note that for solvability of the local problem in each element, the following condition on the trace variable must be satisfied

$$\big\langle \hat{\boldsymbol{u}}_h \cdot \boldsymbol{n}, 1\big\rangle_{\partial\Omega_i} = 0, \quad i = 1, \ldots, \mathtt{n_{el}}. \tag{2.23}$$

Thus, following Nguyen et al. (2011c), the HDG discretization of system (2.21) leads to the following problem: find an approximation $(\boldsymbol{L}_h, \boldsymbol{u}_h, p_h, \hat{\boldsymbol{u}}_h, \boldsymbol{\rho}_h) \in [\widehat{\mathcal{V}}_h]^{d\times d} \times [\widehat{\mathcal{V}}_h]^d \times \widehat{\mathcal{V}}_h \times [\widehat{\Lambda}_h(\boldsymbol{g})]^d \times \mathbb{R}^{\mathtt{n_{el}}}$ such that

$$\left.\begin{aligned}
\big(\boldsymbol{L}_h, \boldsymbol{G}\big)_{\Omega_i} + \big(\boldsymbol{u}_h, \boldsymbol{\nabla}\cdot\boldsymbol{G}\big)_{\Omega_i} - \big\langle \hat{\boldsymbol{u}}_h, \boldsymbol{G}\boldsymbol{n}\big\rangle_{\partial\Omega_i} &= 0,\\
\big(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}\big)_{\Omega_i} + \big(\nu\boldsymbol{L}_h - p_h\mathbf{I} - \boldsymbol{u}_h\otimes\boldsymbol{u}_h, \boldsymbol{\nabla}\boldsymbol{v}\big)_{\Omega_i} &\\
+\big\langle(-\nu\boldsymbol{L}_h + p_h\mathbf{I} + \hat{\boldsymbol{u}}_h\otimes\hat{\boldsymbol{u}}_h)\boldsymbol{n} + \mathbf{S}(\boldsymbol{u}_h - \hat{\boldsymbol{u}}_h), \boldsymbol{v}\big\rangle_{\partial\Omega_i} &= \big(\boldsymbol{b}, \boldsymbol{v}\big)_{\Omega_i}\\
-\big(\boldsymbol{u}_h, \boldsymbol{\nabla} v\big)_{\Omega_i} + \big\langle \boldsymbol{n}\cdot\hat{\boldsymbol{u}}_h, v\big\rangle_{\partial\Omega_i} &= 0,\\
\big\langle p_h, 1\big\rangle_{\partial\Omega_i} &= \rho_{h,i},
\end{aligned}\right\} \tag{2.24a}$$

for $i = 1, \ldots, \mathtt{n_{el}}$, and

$$\begin{aligned}
\sum_i^{\mathtt{n_{el}}} \big\langle(-\nu\boldsymbol{L}_h + p_h\mathbf{I} + \hat{\boldsymbol{u}}_h\otimes\hat{\boldsymbol{u}}_h)\boldsymbol{n} + \mathbf{S}(\boldsymbol{u}_h - \hat{\boldsymbol{u}}_h), \boldsymbol{\mu}\big\rangle_{\partial\Omega_i\setminus\partial\Omega_N}&\\
+ \sum_i^{\mathtt{n_{el}}} \big\langle(-\nu\boldsymbol{L}_h + p_h\mathbf{I})\boldsymbol{n} + \mathbf{S}(\boldsymbol{u}_h - \hat{\boldsymbol{u}}_h), \boldsymbol{\mu}\big\rangle_{\partial\Omega_N} &= 0,\\
\big\langle \boldsymbol{n}\cdot\hat{\boldsymbol{u}}_h, 1\big\rangle_{\partial\Omega_i} = 0, \text{ for } i = 1, \ldots, \mathtt{n_{el}},&
\end{aligned} \tag{2.24b}$$

for all $(\boldsymbol{G}, \boldsymbol{v}, v, \boldsymbol{\mu}) \in [\widehat{\mathcal{V}}_h]^{d\times d} \times [\widehat{\mathcal{V}}_h]^d \times \widehat{\mathcal{V}}_h \times [\widehat{\Lambda}_h(\mathbf{0})]^d$, where $\rho_{h,i} \in \mathbb{R}$ denotes the i-th component of $\boldsymbol{\rho}_h$.

In the HDG local problems (2.24a) the Dirichlet boundary condition (2.22) has been replaced in the boundary integrals. Thus, the continuity of the velocity is weakly imposed by the fact that the trace variable $\hat{\boldsymbol{u}}$ is single valued on each face in the mesh skeleton Γ. In addition, the trace of the normal stress has been replaced in all boundary integrals by the following numerical trace

$$(\nu\hat{\boldsymbol{L}}_h - \hat{p}_h\mathbf{I})\boldsymbol{n} := (\nu\boldsymbol{L}_h - p_h\mathbf{I})\boldsymbol{n} + \mathbf{S}(\hat{\boldsymbol{u}}_h - \boldsymbol{u}_h). \tag{2.25}$$

Matrix $\mathbf{S}$ is the so-called *stabilization tensor*, and may depend on $\boldsymbol{u}_h$ and $\hat{\boldsymbol{u}}_h$, see Nguyen et al. (2011c) for details on the optimal choice. However, a direct dependence of $\mathbf{S}$ on the solution can lead to a cumbersome implementation. For the sake of simplicity, a constant scalar stabilization parameter is chosen in this work, that is $\mathbf{S} = \tau\mathbf{I}$, where τ is a positive constant set to $\tau \approx \max_\Omega(|\boldsymbol{u}|)$, i.e. of the order magnitude of the expected maximum value of the velocity in the domain. This choice provides a simple and effective criterion to tune the stabilization tensor.

The HDG discrete problem (2.24) is a system of Differential Algebraic Equations (DAE) of index 1, that can be efficiently discretized in time with an implicit time integrator, such as backward Euler, a Backward Differentiation Formula (BDF), or a diagonally implicit Runge-Kutta method (DIRK), see Nguyen et al. (2011c). The time discretization of (2.24) leads to a non-linear system of equations – to compute the solution at time t^n from the solution at previous time steps – that can be iteratively solved. Here, the non-linear system has been linearized using the Newton-Raphson method, that is, the generic non-linear term associated to a generic trilinear form is approximated as $c(\boldsymbol{u}^r, \boldsymbol{u}^r; \boldsymbol{v}) \approx c(\boldsymbol{u}^{r-1}, \boldsymbol{u}^r; \boldsymbol{v}) + c(\boldsymbol{u}^r, \boldsymbol{u}^{r-1}; \boldsymbol{v}) - c(\boldsymbol{u}^{r-1}, \boldsymbol{u}^{r-1}; \boldsymbol{v})$, where r is the iteration index, see for instance Nguyen et al. (2011c). In any case, a linear system of equations is to be solved in each iteration of the non-linear solver, to compute a new approximation of the solution at time n. In this linear system, the equations corresponding to (2.24a) can be solved element-by-element to express the solution at each element Ω_i in terms of the trace variable $\hat{\boldsymbol{u}}_h$ and the mean of the pressure on $\partial\Omega_i$, $\rho_{h,i}$. Then, replacing in (2.24b) yields a global system of equations

involving only $\hat{\boldsymbol{u}}_h$ and $\boldsymbol{\rho}_h$, with an important reduction in number of DOF. Further details on the efficient solution of the non-linear DAE (2.24) can be found in Nguyen et al. (2011c). Some details on the implementation of the HDG method for the Navier-Stokes equations are also given in Appendix C.

Also for the Navier-Stokes equations, the particular form of the numerical fluxes used in HDG yields optimal rates of convergence of order $p+1$ in $\mathcal{L}^2$ norm for the three unknowns of the problem $\boldsymbol{L}_h$, $\boldsymbol{u}_h$ and p_h, see Nguyen et al. (2011c). Then, for each element, a new problem is defined element-by-element in the new unknown $\boldsymbol{u}_h^*$, which is the postprocessed solution. The local postprocess proposed by Nguyen et al. (2010) is used here, which requires solving

$$\left.\begin{aligned} -\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\boldsymbol{u}_h^* &= \boldsymbol{\nabla}\cdot\boldsymbol{L}_h && \text{in } \Omega_i \\ -\boldsymbol{\nabla}\boldsymbol{u}_h^*\boldsymbol{n} &= \boldsymbol{L}_h\boldsymbol{n} && \text{on } \partial\Omega_i \\ \int_{\Omega_i} \boldsymbol{u}_h^*\, d\Omega &= \int_{\Omega_i} \boldsymbol{u}_h\, d\Omega \end{aligned}\right\} \text{ for } i = 1,\ldots,\mathtt{n_{el}}.$$

This induces a weak problem in a richer finite dimensional space, that is, find $\boldsymbol{u}_h^* \in [\widehat{\mathcal{V}}_h^*]^d$ such that

$$\big(\boldsymbol{\nabla}\boldsymbol{u}_h^*, \boldsymbol{\nabla}\boldsymbol{v}\big)_{\Omega_i} = \big(\boldsymbol{L}_h, \boldsymbol{\nabla}\boldsymbol{v}\big)_{\Omega_i} \text{ and } \big(\boldsymbol{u}_h^*, 1\big)_{\Omega_i} = \big(\boldsymbol{u}_h, 1\big)_{\Omega_i}$$

for all $\boldsymbol{v} \in [\widehat{\mathcal{V}}_h^*]^d$ and $i = 1,\ldots,\mathtt{n_{el}}$, where $\widehat{\mathcal{V}}_h^*$ must be a bigger space than $\widehat{\mathcal{V}}_h$. As for the Laplace equation in Section 2.1, with one degree more in the element-by-element polynomial approximation, i.e. $\widehat{\mathcal{V}}_h^* = \big\{v \in \mathcal{L}^2(\Omega) : v|_{\Omega_i} \in \mathcal{P}^{p+1}(\Omega_i), \text{ for } i = 1,\ldots,\mathtt{n_{el}}\big\}$, $\boldsymbol{u}_h^*$ converges asymptotically at a rate of $p+2$ in the $\mathcal{L}^2$ norm, see Nguyen et al. (2010).

Note that the post-process solution is not required to be computed at each time step, but only when an improved solution is needed.

2.3.2 NACA 0012 airfoil

The goal of this numerical test is to compare the performance of high-order elements ($p > 2$) and low-order elements in terms of computational efficiency, similarly to

what was done in Section 2.2. The test proposed is the numerical evaluation of the aerodynamics characteristics of a symmetric NACA 0012 airfoil, in laminar flow regime, at Re=10 000 and angle of incidence $\alpha = 2°$.

The analytical expression considered for the parametrization of the upper part of the symmetric NACA airfoil is

$$y = \frac{t_k}{0.2}(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4), \text{for } x \in [0, 1],$$

where $t_k = 0.12$ is the maximum thickness of the airfoil. This formula provides a zero thickness airfoil at the trailing edge, which is usually preferred for numerical simulations. A practical procedure for obtaining the computational mesh is illustrated by Giorgiani et al. (2013a).

Four nested meshes of polynomial order $p = 1, 2, 4, 8$ are considered, with the same number of nodes in the domain, see Figure 2.16. Similarly to Nguyen et al. (2011c), the meshes are refined around the airfoil to capture the boundary layer, and also at the leading and trailing edge.

Reference values of the aerodynamic coefficients are obtained using a reference solution, with element size similar to the $p = 2$ computation and $p = 10$. Since the output involves quantities that are not improved by the HDG superconvergent postprocess (that is, the pressure and the velocity gradient), no postprocessed solution is considered in this case. Results of the simulation are summarized in Table 2.5. The error on the aerodynamics coefficients for the four computations is displayed together with the computational cost in terms of DOF, that is, the dimension of the linear system of the problem, and the runtime. In this case the time to perform one Newton-Raphson iteration is considered. As expected, increasing the degree of the approximation the number of DOF of the computation decreases. This is due to the hybridization technique that reduces the coupled unknowns to those of the trace of the velocity, defined only on the boundary of the elements. Also the number of pressure unknowns in the linear system are reduced, being equal to the number of elements. As a consequence, the runtime also decreases increasing the polynomial

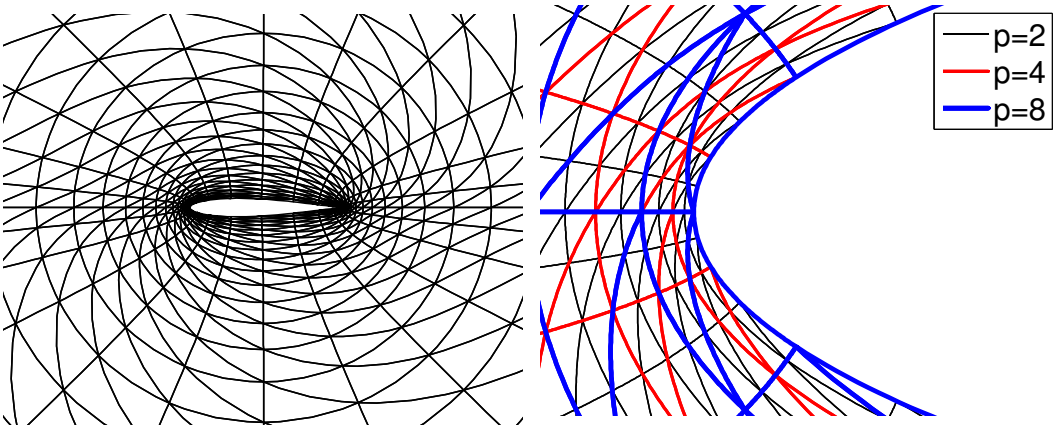


Figure 2.16: NACA 0012 airfoil: mesh $p = 4$ (left) and zoom in the leading edge for the nested meshes $p = 2, 4, 8$ (right).

Table 2.5: NACA 0012 airfoil at Re=10 000 and $\alpha = 2°$: performance comparison between low-order and high-order elements with HDG.

	Comp. cost		**Error %**		
	nDOF	**runtime [s]**	**Lift**	**Drag**	**Moment**
$p = 1$	128832	12.9	103.3	1.1	1.2
$p = 2$	45936	4.5	13.3	0.7	0.06
$p = 4$	18312	3.6	1.0	0.1	0.42
$p = 8$	7956	3.1	18.1	1.4e-3	0.47

degree. Very good accuracy is obtained with the $p = 4$ computation, outperforming linear and quadratic elements. An acceptable accuracy is also obtained with the $p = 8$ computation, despite the very coarse discretization. Note, in Figure 2.17, the comparison between the element size of the $p = 8$ computation in the boundary layer and the velocity profile. Even with this very coarse discretization, a good estimation of the aerodynamic coefficients is obtained. Also the detachment point of the boundary layer, depicted in Figure 2.18, is very well captured with large $p = 8$ elements.

In conclusion, HDG and high-order elements reveals to be a powerful binomial also for fluid problems. This encourages the development of the p-adaptive technique in the next chapter to exploit to the maximum this combination.

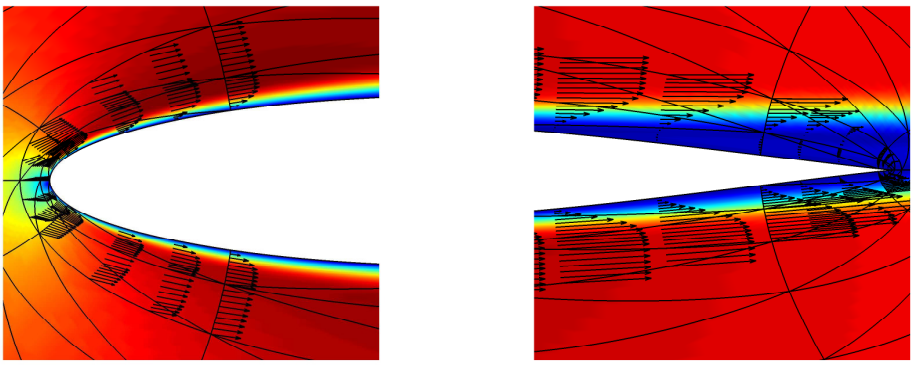


Figure 2.17: NACA 0012 airfoil at Re=10 000 and $\alpha = 2°$: velocity field around the leading edge and the trailing edge at solved with a coarse $p = 8$ mesh. The velocity vector are scaled with a factor of 50 in the leading edge and 10 in the trailing edge, to visualize the start of the recirculation.

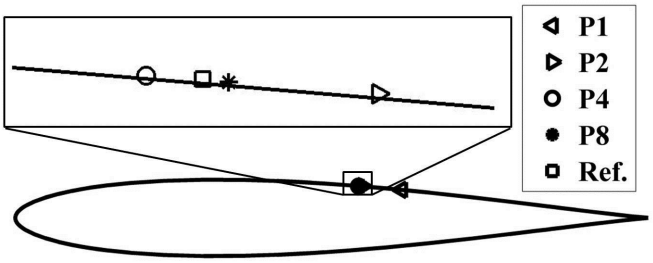


Figure 2.18: NACA 0012 airfoil at Re=10 000 and $\alpha = 2°$: detachment point of the boundary layer as captured by the four simulations, and comparison with the position obtained with the reference solution.

2.4 Conclusions

The HDG method applied to the solution of 2D scattering problems in unbounded domains. Moreover, non-constant coefficients are considered in the wave propagation domain. This is typical of the Mild Slope equation for modeling the propagation of sea waves in harbors.

For this wave propagation problem, the HDG method is compared with the standard continuous Galerkin (CG) method and the Compact Discontinuous Galerkin

(CDG) method, as a representative DG method with compact stencil. The efficiencies of the three methods are evaluated in terms of accuracy, dimension of the global linear system (number of DOF) and runtime. HDG with an element-by-element decoupling of the interior nodes, induces an important reduction in number of DOF for high-order elements. Analogously to static condensation in CG, the actual DOF in the HDG method correspond to nodes on the element boundaries. Consequently, HDG exhibits an efficiency comparable to CG, providing similar levels of accuracy for the same computational cost, and clearly outperforming CDG. However, HDG is always more expensive but more accurate than CG on the same discretization. Nevertheless, as the degree of the approximation increases these differences in runtime and accuracy between HDG and CG become smaller.

At the same time, high-order approximations are compared with low-order approaches for a wide range of wavelengths. High-order elements outperform low-order elements in terms of efficiency of the computation, since they require less degrees of freedom for a given precision and also less computational time, both for high-accuracy and low-accuracy. This is also the case in examples with singularities and holds both for continuous and discontinuous methods.

The HDG method is also applied to the solution of the Navier-Stokes equations. The computational efficiency of high-order and low-order approximations are compared in a 2D numerical test regarding the evaluation of the aerodynamic characteristics of a NACA 0012 airfoil in laminar regime. High-order elements again provide high accuracy computations outperforming low-order approaches both from the point of view of the linear system dimension and the runtime.

Chapter 3

Approximation degree adaptive strategies for HDG

The previous chapter highlighted the effectiveness of high-order HDG computations for wave propagation problems and fluid problems compared to low-order approximations. High-order elements provide better accuracy for the same computational cost, or require less computational cost for a desired accuracy level. However, the discontinuous character of the solution in the HDG method opens the path to a further optimization: p-variable computations, that is, increasing the polynomial degree only where more accuracy is needed.

Typically, adaptive procedures start from a discretization designed to exploit the *a priori* knowledge of the solution (for example, the presence of boundary layers) and to accurately represent the geometry of the domain. Then, adaptivity automatically adjusts the elements locally, relying on error indicators or error estimators computed from the solution itself, in order to provide the necessary spatial resolution and to accurately capture the solution as it evolves.

In this chapter, a p-adaptive HDG method for wave propagation problems and incompressible flows problems is proposed. Exploiting the superconvergent post-process of HDG, a simple and reliable error estimator is derived, that involves only element-by-element computations. Thus, as shown in the numerical examples of the previous

chapter, the computational cost of the error estimator is significantly smaller than the cost of computing the solution. The error estimator drives the automatic update of the approximation degree in each element, which is aimed at obtaining a uniform error distribution below a user defined tolerance. No mesh topology changes are introduced, thus the adaptive process is also very fast compared with the computation of the solution. Despite the simplicity of the proposed error estimator and the adaptive technique, high accuracy computations are obtained, with sharp resolution of the key features of the flow and accurate evaluation of the fluid-dynamic forces. Moreover, the p-adaptive algorithm greatly simplify the design of the initial computational mesh, reducing for example the need of highly distorted elements for capturing the boundary layer. An important reduction in computational cost is shown in comparison to p-uniform computations. Applications to steady-state and transient problems are presented.

3.1 Error estimation and p-adaptive strategy

The use of *a posteriori* computable error estimates to drive an automatic adaptive process is nowadays a common practice to control the accuracy of the computation and to ensure the quality of the FE solution, see for example Ainsworth and Oden (2000) . Two different approaches can be used for assessing the error: error indicators or error estimators, see Huerta et al. (1999) for a general discussion. Error indicators are cheap to evaluate but, in general, are designed for a specific problem and do not provide error bounds. Error estimators are more accurate and general, can be used in linear and nonlinear problems, see for instance Díez et al. (2000), and can even produce bounds of the exact solution as proposed by Sauer-Budge et al. (2004) and Parés et al. (2006, 2008, 2009). But, as noted by Díez et al. (2007), a posteriori error estimators have a non trivial computational overhead when recovery techniques are used, or by Ladevèze and Pelle (2005) when equilibrated fluxes must be computed , and also in flux-free implementations (Parés et al., 2006). Therefore, the superconvergent post-process inherent in HDG is a important asset, which can be readily employed to

estimate the error.

Both for wave and fluid problems, a measure of the elemental error in the domain of interest $\Omega_{\texttt{int}}$ is prescribed below a user-given tolerance, ε, namely

$$\max\{E_i|\Omega_i \in \Omega_{\texttt{int}}\} = E_{\texttt{int}} \leq \varepsilon \tag{3.1}$$

with E_i being the error in the element Ω_i.

Instead of point-wise errors, the $\mathcal{L}^2$ norm is considered in each element, that is

$$E_i^2 = \frac{1}{A_i}\int_{\Omega_i} |\circledcirc^* - \circledcirc|^2 \, d\Omega, \tag{3.2}$$

where $\circledcirc^*$ is the post-processed solution and A_i the measure of element Ω_i. For wave problems, $\circledcirc = H$, that is, the wave amplification factor defined in (2.19), while for fluid problems $\circledcirc = \boldsymbol{u}$, that is, the velocity. Note that the evaluation of the improved solution $\circledcirc^*$ requires an inexpensive element-by-element computation, see the numerical tests of the previous chapter, that is actually done only when the error estimator is to be evaluated. Thanks to the superconvergent properties of the HDG method, the error estimator defined in (3.2) is an asymptotically exact estimator, both for uniform and variable degree distributions.

The adaptive strategy proposed here is aimed at obtaining a uniform error distribution and complying with the condition (3.1). The adaptive process is based on estimating the variation in the approximation degree p_{Ω_i}, for each element in the computational domain, to reach the desired accuracy in the area of interest. Given an approximation obtained with a p_{Ω_i}-map of interpolation degrees, a degree increment, Δp_{Ω_i}, is evaluated for each element i.e. $i = 1, \ldots, \texttt{n}_{\texttt{el}}$. Inspired from Remacle et al. (2003), the p_{Ω_i} variation in each element is computed as

$$\Delta p_{\Omega_i} = \left\lceil \log_b(E_i/\epsilon_i) \right\rceil, \tag{3.3}$$

where $\lceil\cdot\rceil$ denotes the *ceiling* function, that is, a function that maps a real number to the smallest following integer. The elemental tolerance ϵ_i usually corresponds to the

desired tolerance ε, but can also be set differently element-by-element, to intensify the refinement in a particular area of the domain, see for instance Giorgiani et al. (2013b).

Similarly to the *adaptation aggressiveness* parameter by Fidkowski and Darmofal (2007), the logarithm base b controls the behavior of the adaptive scheme: for a fixed elemental error ratio E_i/ϵ_i, increasing b has the obvious effect of decreasing Δp_{Ω_i}. Thus, small values of b yield drastic variations in the polynomial degree in the elements. This may reduce the number of iterations until convergence but it can also produce oscillatory p_{Ω_i}-map around the optimal one. On the contrary, large values of b yield slow variations on the p-map. The number of iterations to determine the optimal distribution of polynomial degrees is consequently increased but it converges to the optimal map without undesired oscillations. The optimal value of the parameter b depends on the tolerance ε required in the computation and on the specific problem at hand. However, for the examples studied in this work reasonable results are obtained with $10 \leq b \leq 100$. This bounds correspond to an increment $\Delta p = 1$ for $E_i/\epsilon_i = 10$ and 100 respectively.

In addition, an upper and lower bound for the approximation degree is usually defined, namely $p^- \leq p_{\Omega_i} \leq p^+$. On one hand, in case of fluid problems, the lower bound is only used to guarantee a correct representation of the exact geometry with isoparametric elements. The upper bound, which limits the maximum polynomial degree in the mesh, may have an influence in the results if the elements are excessively large for the desired tolerance. An *hp* adaptive strategy would be necessary in this case to obtain convergence. However, the *a priori* information on the solution allows to distribute the element sizes in such a way that this problem does not occur, as it will be shown in the numerical examples.

On the other hand, in case of wave problems, the lower bound should also prescribe a minimum number of nodes per wavelength. In fact, defining a suitable number of elements per wavelength is not a trivial task. A "rule of thumb" proposed by Thompson and Pinsky (1994) and Zienkiewicz et al. (2005) usually suggests 10 linear elements per wavelength, and, obviously, less than 4 linear elements is impossible.

To roughly estimate p^- and p^+, assume that high-order elements require the same number of nodes per wavelength than linear ones, which is on the safe side because as noted by Thompson and Pinsky (1994) high-order elements require less nodes per wavelength. Since the number of nodes per one wavelength is $1 + p(2\pi)/kh$ where h is the characteristic element size, a reasonable upper bound for the degree of approximation can be estimated with $p^+ \approx 10kh/2\pi$ and an absolute lower bound by $p^- \approx 4kh/2\pi$. Note however that results are very insensitive to the prescribed lower bound because the adaption process puts enough nodes even if a lower value for p^- is prescribed. On the contrary, the upper bound may prevent convergence of the adaptive process if the elements are excessively large (for the desired tolerance and the solution wavelength). Also in this case an *hp* adaptive strategy would be necessary to achieve convergence. For the example presented, h refinement was unnecessary even for very large elements (a characteristic element size equal to incoming wavelength, i.e. $kh \approx 2\pi$) provided engineering accuracy was imposed (i.e. approximately two significant digits) and $p^+ = 10$, as suggested from the previous safe estimate.

Two adaptive procedures are considered, one for steady state solutions and the other for transient solutions, see also Wu et al. (1990) and Oñate et al. (2006). Both of them start from a mesh with uniform degree distribution, that is, $p_{\Omega_i} = p^-$ for $i = 1, \ldots, \mathtt{n_{el}}$.

The adaptive procedure for steady-state solutions concerns the solution of wave problems described by system (2.13) and also the Navier-Stokes equation in the case that a steady-state is sought. In this case, each iteration of the adaptive process consists on computing the solution of (2.13) or the the steady-state solution of (2.24), evaluating the error estimate (3.2), updating the degree map with (3.3) and the bounds $p^- \leq p_{\Omega_i} \leq p^+$, projecting the solution onto the new computational mesh and restart the computation.

Remark 7. *Note that, for the Navier-Stokes equations, for the second and subsequent iterations of the adaptive process, a good initial approximation of the solution, $\boldsymbol{u}_0$, is available from the previous iteration. Thus, few time steps, with large step size, reach the steady state solution with low computational cost. For example, in the first two*

numerical examples presented in Section 3.3, an infinite time step is used for second and subsequent iterations of the adaptive process, which is equivalent to solving the stationary version of (2.20)*, reducing the computational cost of each adaptive iteration to the Newton-Raphson solver iterations.*

The iterative adaptation scheme stops when the prescribed precision ε is attained in the domain. Failure to converge to the desired accuracy is assumed when the percentage of elements changing their degree in two successive iterations is lower than a given value, and the target ε is not accomplished.

For purely transient problems, error estimation and consequent degree adaptation is repeatedly performed after a fixed number of time steps. The solution is then projected onto the new mesh and the simulation is continued.

The flow charts of the two adaptive strategies are shown in Figure 3.1.

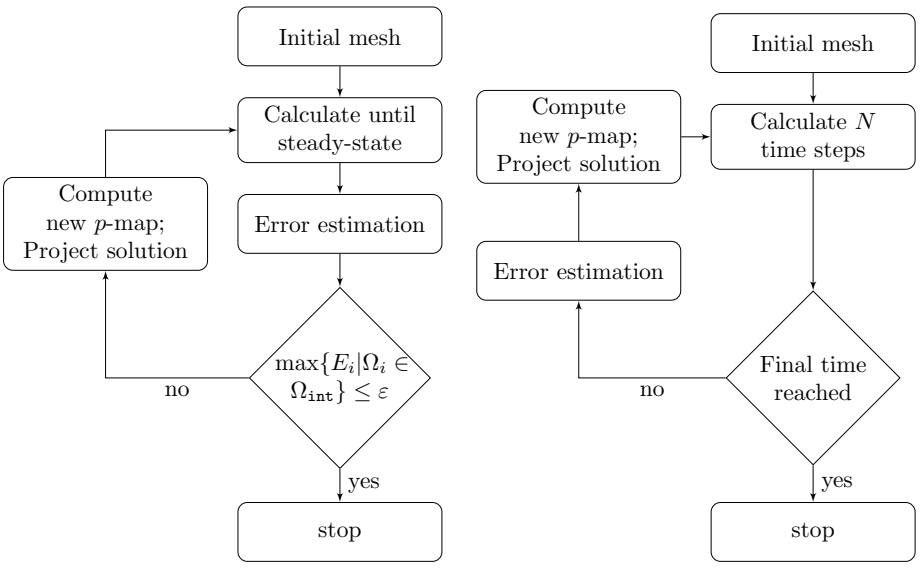


Figure 3.1: Adaptive strategy for steady-state solutions (left) and for transient solutions (right).

3.2 P-adaptive HDG for wave problems: numerical tests

The performance of the proposed adaptive HDG method is studied in real cases of sea wave propagation in harbors. The model problem is the Mild Slope equation as described in Section 2.2. The first application is a medium size harbor (Mataro's harbor) and the second is a large one (Barcelona' harbor). HDG adaptive solutions are compared with high-order CG solutions, assessing the efficiency of the method in terms of number of DOF of the linear system for the same accuracy in the area of interest. Obviously, no analytical solution are available for these examples. The errors are computed using the proposed error estimator for HDG. For CG the reference solution is computed on a $p+1$ mesh. Note that the computational overhead for CG is non-negligible and it is not introduced in the comparisons. The efficiency of the HDG error estimator is also evaluated in the two test cases. To do so, a reference solution is evaluated in each case with a high-order CG approximation on an h-refined mesh. Since discretization errors are of concern here, the PML region is kept unchanged in every computation, also for the reference computation.

Mataró harbor

Figure 3.2 (left) depicts the first problem statement, note that the PML region is highlighted. The docking area is considered as the area of interest and it is shown in Figure 3.2. The physical boundaries are also indicated, they are modeled as absorbing boundaries with $\alpha = 0.02$ for dikes, $\alpha = 0.4$ for breakwaters, and $\alpha = 0.7$ for beaches. The incident wave direction is 10° from the x-axis, which should induce strong agitation in the interior. The wave period is 5s, corresponding to short waves with a maximum value of the wavelength of about 40m in the PML region, and a minimum value of 25m in the interior of the harbor and close to the beach. Real bottom depth has been used everywhere except in the PML region where constant depth is imposed. A high-order meshing software developed by Sarrate et al. (2011) is used to generate an unstructured triangular mesh with a uniform characteristic size

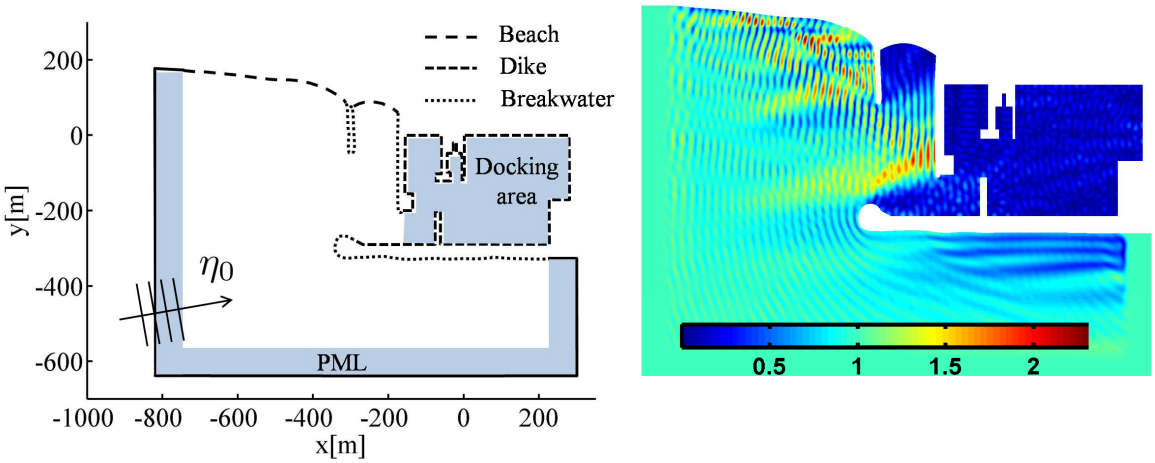


Figure 3.2: Mataro's harbor: problem statement (left) and wave amplification factor solution (right)

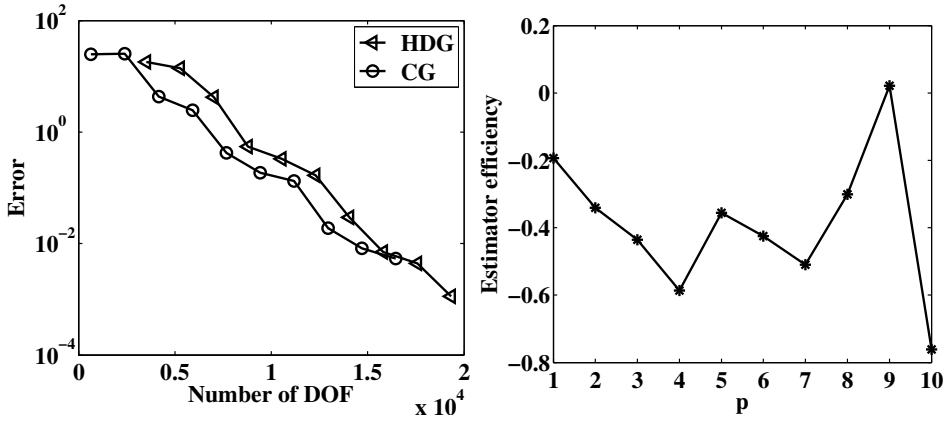


Figure 3.3: Mataro's harbor: error vs. DOF for non-adaptive, i.e. uniform p (left), for HDG and CG; efficiency of the error estimator in HDG (right).

$h \approx 40$m everywhere in domain, except for the interior of the harbor where the mesh size is adapted locally to capture all the relevant geometrical features of the docking area.

The wave amplification factor in the domain is also displayed in Figure 3.2 (right). Note the increase of the wave-height (amplification factors larger than two) due to the bathymetry in the harbor entry and on the beach outside the harbor. Figure 3.3 (left) shows the convergence of the maximum elemental wave-height $\mathcal{L}^2$ error in

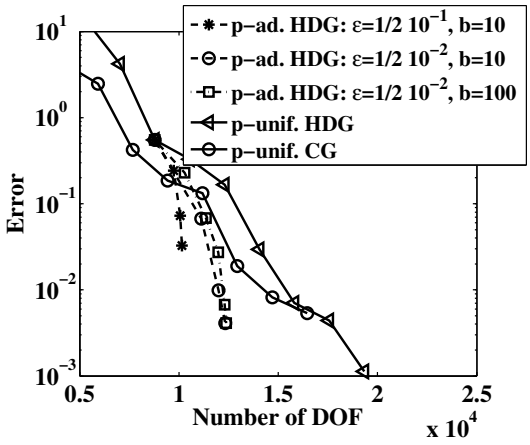


Figure 3.4: Mataro's harbor: zoomed convergence curves for p-uniform CG and HDG and p-adaptive HDG computations.

the area of interest, i.e. $E_{\mathtt{int}}$ see (3.1), versus the number of DOF. Each mark of the convergence curves corresponds to a simulation with uniform p; along each curve p increases from 1 to 10 on the same mesh for HDG and CG. As noted in Section 2.2, CG is always more efficient (in terms of DOF) than non-adaptive DG methods for a similar accuracy. However, as expected, the number of DOF for HDG is close to CG as p increases.

In Figure 3.3 (right), the efficiency of the error estimator is evaluated for each computation. The estimator efficiency is defined as

$$E_{\mathtt{int}} / \max\{\tilde{E}_i | \Omega_i \subset \Omega_{\mathtt{int}}\} - 1, \tag{3.4}$$

where $\tilde{E}_i$ is the "real" elemental error, that is, the error computed with the reference solution (an overkilled h-refined CG solution). The graphic shows that, in this case, the estimator is almost always underestimating $E_{\mathtt{int}}$. Moreover, the efficiency is, for most of the range of p larger than 1/2. Comparisons with three different strategies of p-adaptive HDG solutions are depicted in Figure 3.4. To better appreciate the improvements, a zoom, with respect to the left figure, is shown and only the HDG and CG methods are plotted. Without recourse to computational efficiency of the different schemes, p-adaptive HDG requires less DOF for a given accuracy compared with

uniform CG. No CG p adaptivity is compared because it is clearly more cumbersome to implement than in DG. This figure illustrates the convenience of the p-adaptive technique, which allows to reduce considerably the DOF of the computation, providing better performance than uniform CG approximations. This test also shows the reliability of the error estimator in the task of driving an adaptive process.

This results are obtained with the methodology proposed in Section 3.1 and imposing two different tolerances $\varepsilon = 0.5\,10^{-1}$ and $\varepsilon = 0.5\,10^{-2}$ (i.e. one and two significant digits, see Higham (2002), which cover the usual engineering accuracy needs). The targeted elemental tolerance was prescribed imposing $\epsilon_i = 0.5\varepsilon$ for $\Omega_i \subset \Omega_{\texttt{int}}$ and $\epsilon_i = 2\varepsilon$ for $\Omega_i \subset \Omega \setminus \Omega_{\texttt{int}}$. Since an almost uniform element size was is imposed equal to the incident wave length, i.e. $h \approx 2\pi/k_0 = 40$m, the lower bound for the approximation degree is $p^- = 4$ and the upper bound $p^+ = 10$ is never a restriction. Two values of the parameter b, see (3.3), are also tested for the case $\varepsilon = 0.5\,10^{-2}$, $b = 10$ and $b = 100$, leading to almost the same converging point, the first one with three iterations and the second one with five iterations. As expected lower values of b converge faster. However, as noted earlier, if b is further reduced convergence can be lost, this is the case for $b = 2$. Figure 3.5 (left) depicts the p_{Ω_i}-map for $\varepsilon = 0.5\,10^{-2}$. Note that high values of p are required in the area of interest but also outside $\Omega_{\texttt{int}}$ where interactions and water depth have an influence in the accuracy in the area of interest. The elemental errors in the area of interest are shown in Figure 3.5 (right). Obviously, the maximum error is below the prescribed value ε, and the majority of the element stay in a range of error comprised in one order of magnitude. Thus, a reasonably uniform error distribution is achieved.

Barcelona harbor

The same computational mesh of Section 2.2.3 is used in this example. Figure 3.6 shows a map of the bottom depth. It coincides with actual data except for the vertical left and right boundaries where it is smoothed to simplify the imposition of the PML conditions. Figure 3.7 (left) plots $E_{\texttt{int}}$ (the maximum elemental $\mathcal{L}^2$ error in the wave amplification factor for the area of interest) versus the number of DOF of

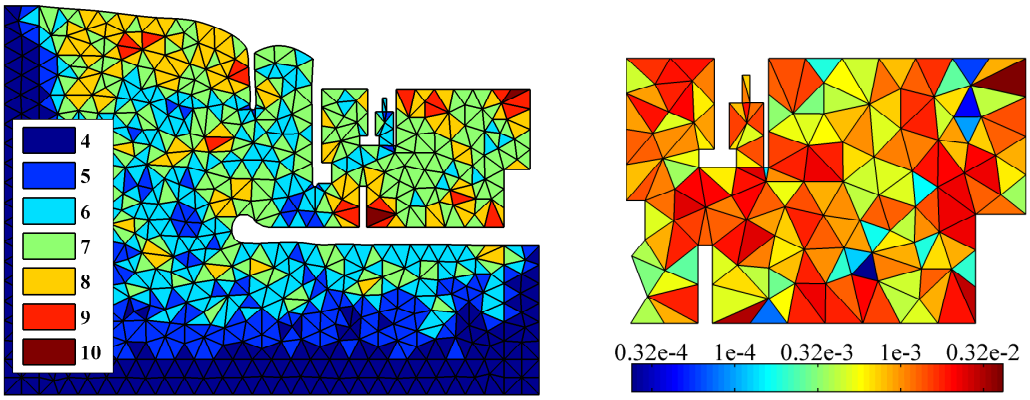


Figure 3.5: Mataro's harbor: adaptive results for $\varepsilon = 0.5\,10^{-2}$, p-map (left) and error map in area of interest (right)

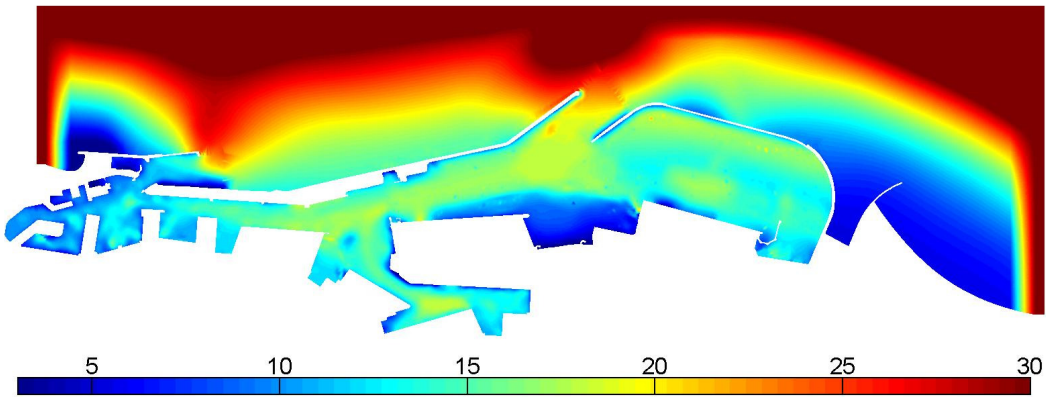


Figure 3.6: Barcelona harbor: bottom depth

the computation (solid line). Markers are plotted for $p = 5, \ldots, 12$ since p uniform HDG computations are performed. This curve (solid line), based on the estimate of the post-processed solution, can be compared with the "exact" error (dashed line) where the reference solution is obtained with CG, the same mesh, and $p = 20$. Note the good agreement between the estimate and the "exact" error. To further study the estimate Figure 3.7 (right) shows the efficiency of the error estimator as a function of p for these computations. Again, very good efficiencies are obtained.

For the HDG p-adaptive simulations, the target accuracy is set to $\varepsilon = 10^{-2}$ and $\varepsilon = 0.5\,10^{-2}$, the lower bound for the order of the approximation is $p^- = 5$ and again,

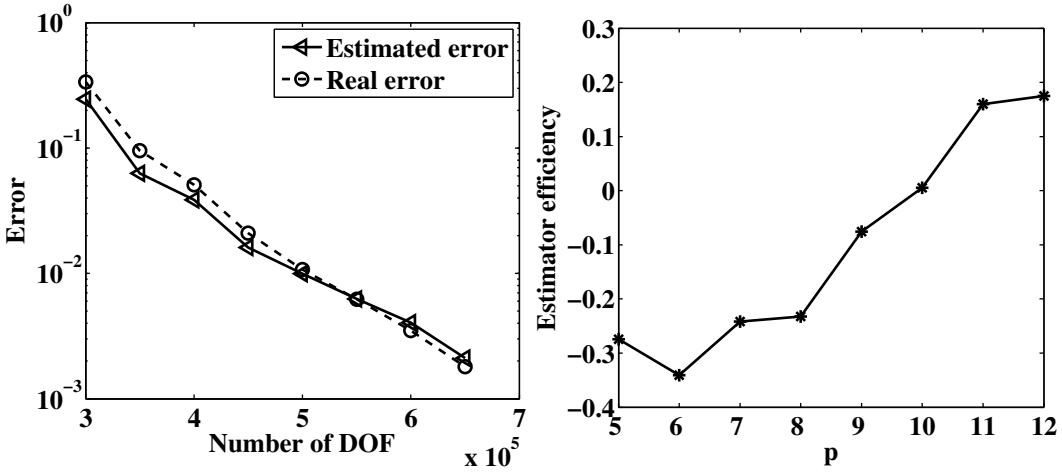


Figure 3.7: Barcelona harbor: analysis of the error estimate for uniform p HDG computations. Comparison between $E_{\texttt{int}}$ and the error computed with a reference solution for $p = 5, \ldots, 12$ (left) and efficiency of the error estimator (right).

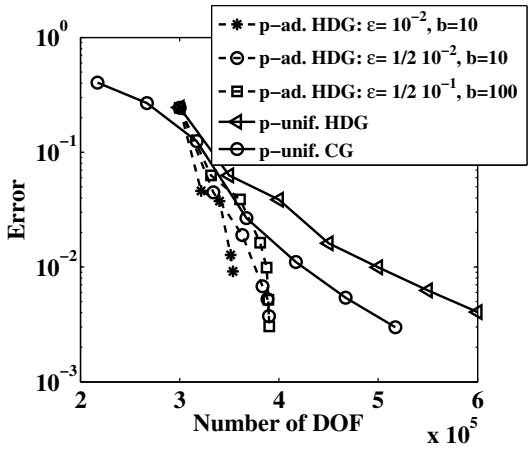


Figure 3.8: Barcelona harbor: convergence comparison between p-uniform HDG and CG for $p = 5, \ldots, 11$ and three HDG p-adaptive computations.

there is no need to limit the polynomial degree with an upper cap. To drive the adaptive process, the element target error is set as in the previous example, that is $\epsilon_i = 0.5\varepsilon$ for $\Omega_i \subset \Omega_{\texttt{int}}$ and $\epsilon_i = 2\varepsilon$ for $\Omega_i \subset \Omega \setminus \Omega_{\texttt{int}}$. Figure 3.8 shows convergence for p-adaptive HDG and p-uniform HDG and CG computations. For $\varepsilon = 10^{-2}$, only $b = 10$ is considered and convergence is obtained with four iterations of the adaptive process. For $\varepsilon = 0.5\,10^{-2}$, two values for b are compared: $b = 10$ and $b = 100$. As

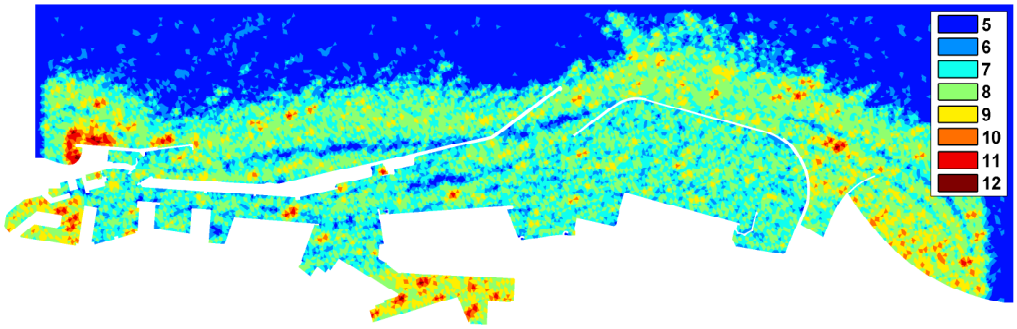


Figure 3.9: Barcelona harbor: map of the approximation degree p at convergence, for the case $\varepsilon = 0.5\,10^{-2}$ and $b = 10$.

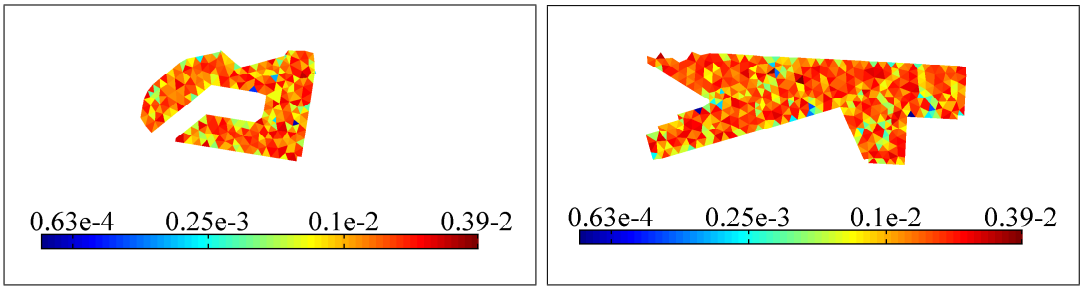


Figure 3.10: Barcelona harbor: error map in the docking areas.

expected less iterations (five) are necessary for the lower b while six iterations are required for $b = 100$. Nevertheless, they both reach similar converged solutions. Note that every adaptive computation induces a significative reduction in number of DOF compared to p-uniform CG and HDG computations. As noted earlier if b is too small, viz. $b = 2$, oscillatory no convergent p-maps can be obtained.

Figure 3.9 shows the distribution of order p for $\varepsilon = 0.5\,10^{-2}$ and $b = 10$. Note that the maximum p is only imposed in small areas. Moreover, there is a certain correspondence between high p and shallow waters where the wavelength is reduced, see Figure 3.6. As expected the p-adaptive technique is automatically putting the necessary nodes per wavelength in each part of the domain.

Finally, the elemental errors in the docking area for the case $\varepsilon = 0.5\,10^{-2}$ and $b = 10$ are plotted in Figure 3.10 with an almost uniform error distribution in the zone of interest.

Table 3.1: Barcelona harbor: runtime, error and DOF comparison for increasing p-uniform CG and p-adaptive HDG, with a desired tolerance of $\varepsilon = 0.5\,10^{-2}$ and $b = 10$.

	iteration	0	1	2	3	4
CG	p	5	7	9	10	11
	error	0.40	0.12	$0.11\,10^{-1}$	$0.54\,10^{-2}$	$0.30\,10^{-2}$
	DOF	217 198	317 198	417 198	467 198	517 198
	runtime solve [s]	18	38	130	168	317
HDG	p or its range	5	$5,\dots,8$	$5,\dots,10$	$5,\dots,11$	$5,\dots,12$
	error	0.24	$0.45\,10^{-1}$	$0.19\,10^{-1}$	$0.64\,10^{-2}$	$0.36\,10^{-2}$
	DOF	300 000	333 832	362 028	380 842	384 951
	runtime solve [s]	64	82	100	115	122
	runtime err. est.[s]	17	27	33	40	42

Table 3.1 compares runtime, error and DOF in a p-refinement process for p-uniform CG and p-variable HDG to demonstrate the advantages of the p-adaptive HDG strategy. The desired accuracy is $\varepsilon = 0.5\,10^{-2}$, $\epsilon_i = 0.5\varepsilon$ for $\Omega_i \subset \Omega_{\texttt{int}}$ and $\epsilon_i = 2\varepsilon$ for $\Omega_i \subset \Omega \setminus \Omega_{\texttt{int}}$, and $b = 10$. CG computations use uniform degree p. The error can be estimated with any a posteriori error estimator. Here, for simplicity a new computation with uniform $p+1$ is done and, obviously, the runtime indicated in Table 3.1 for CG only accounts for the first solve with the approximation of order p. For each iteration, the uniform CG degree is updated as the maximum degree in the p-map obtained using (3.3). The table presents the degree p for each CG computation, its corresponding maximum estimated error in the area of interest, the number of DOF and the runtime. Four iterations of this process are needed to attain the desired precision. Note that, respect to Table 2.2, slightly different error are obtained due to the fact that here the reference solution is considered as the $p+1$ computation.

HDG uses the proposed methodology described earlier. The results are also shown in Table 3.1. The first iteration uses the same mesh and $p = 5$ as the first CG iteration. As expected HDG is more computationally expensive than CG but provides better accuracy for the same discretization. Now, however, the post-processed approximation provides a reasonable error estimate with a reduced overhead (one third of the coarse computation), that is also indicated in the table. The following iterations have an adapted p distribution. The table shows the range of p, the maximum $\mathcal{L}_2$ error

in the area of interest, the number of DOF, the run-time for the solution and the run-time for the HDG error estimate. Four iterations are also required to attain the desired accuracy and the successive iterations show how the proposed p-adaptive algorithm improves the efficiency of the HDG computations. Note that the sum over the iterations of runtimes is 483s for HDG and 671s for CG and that any error estimator for CG will be more computationally expensive than the post-process of HDG.

3.3 P-adaptive HDG for fluid problems: numerical tests

In this section, the proposed p-adaptive technique is applied to the solution of the Navier-Stokes equations. The performance of the adaptive technique is tested through several numerical examples. In the first one, the Wang flow in a square domain is considered. The analytical solution allows, in this example, to compare estimated errors and exact errors.

The methodology is then applied to problem of engineering interest such as the evaluation of the aerodynamic characteristics of a NACA 0012 airfoil and the computation of the oscillating drag force on a shedding cylinder.

Wang flow

The first example is a closed form solution of the incompressible Navier-Stokes equations introduced by Wang (1991). The computational domain is $\Omega \in]-0.5, 0.5[\times]0, 1[$, and the analytical solution for the velocity is

$$\boldsymbol{u} = \begin{pmatrix} 2ay - b\lambda \exp(-\lambda y)\cos(\lambda x) \\ b\exp(-\lambda y)\sin(\lambda x)\lambda \end{pmatrix}. \tag{3.5}$$

The coefficients in (3.5) are set to $a = 1$, $b = 1$ and $\lambda = 10$, which leads to a concentration of the variation of the velocity and pressure profile to the bottom of

the computational domain. Figure 3.11 shows the streamlines (left), the pressure profile at $x = 0$ (right) and the velocity profile at the bottom boundary (bottom). No

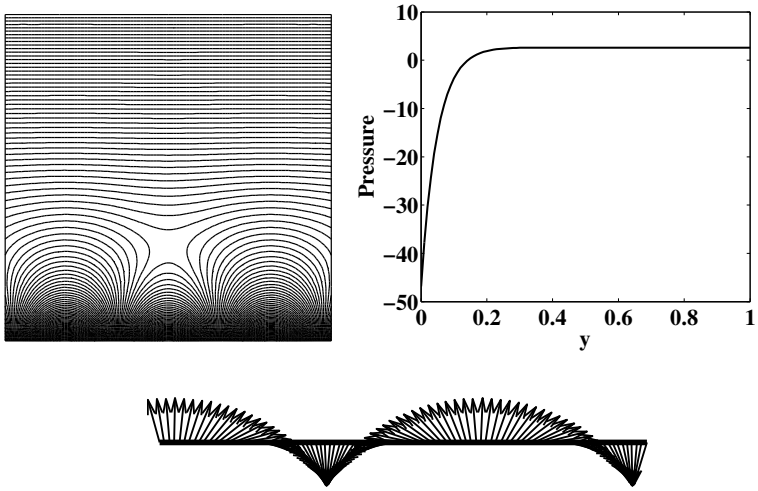


Figure 3.11: Wang flow problem: streamlines (left), pressure profile (right) and velocity profile in the bottom boundary (bottom).

Neumann boundary conditions are imposed in this problem, thus the global problem for pressure is closed setting its average in the domain to zero, i.e. $\left(p_h^n, 1\right)_\Omega = 0$ is added to system (2.24b).

To evaluate the accuracy of the error estimator, the estimated and the exact are compared for a structured triangular mesh with element size $h = 0.1$ and uniform degree p, for $p = 1 \ldots 8$. Figure 3.12 (left) shows the comparison between the estimated and exact maximum $\mathcal{L}^2$ elemental error, exhibiting an excellent agreement. The right figure depicts the estimator efficiency defined by Equation 3.4, where $\Omega_{\texttt{int}} = \Omega$ and in this case $\tilde{E}_i$ is the elemental error computed with the analytical solution. The maximum elemental error is always estimated with an accuracy greater than 95%, demonstrating the excellent performance of the proposed estimator.

Figure 3.13 shows the evolution of the maximum elemental $\mathcal{L}^2$ error for an adaptive computation with tolerance $\varepsilon = 10^{-8}$, uniform elemental tolerance $\epsilon_i = \varepsilon$ for $i = 1, \ldots, \texttt{n}_{\texttt{el}}$, and adaption aggressiveness parameter $b = 100$. Starting from a uni-

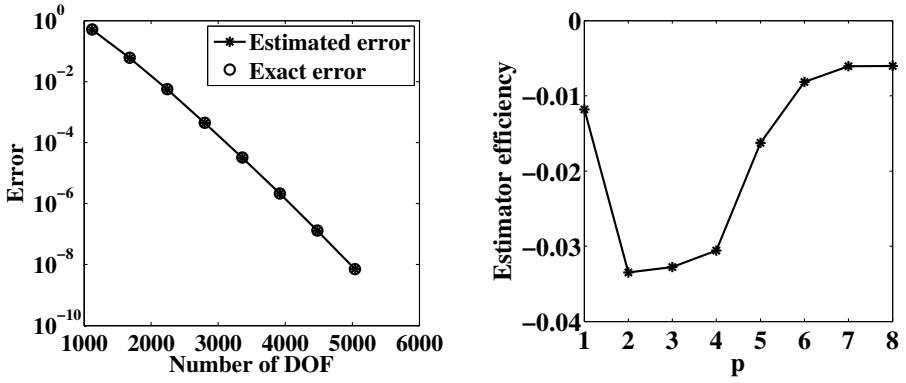


Figure 3.12: Wang flow problem: uniform p computations, $p = 1 \dots 8$. Comparison of the estimated and exact maximum $\mathcal{L}^2$ elemental error (left) and error estimator efficiency (right).

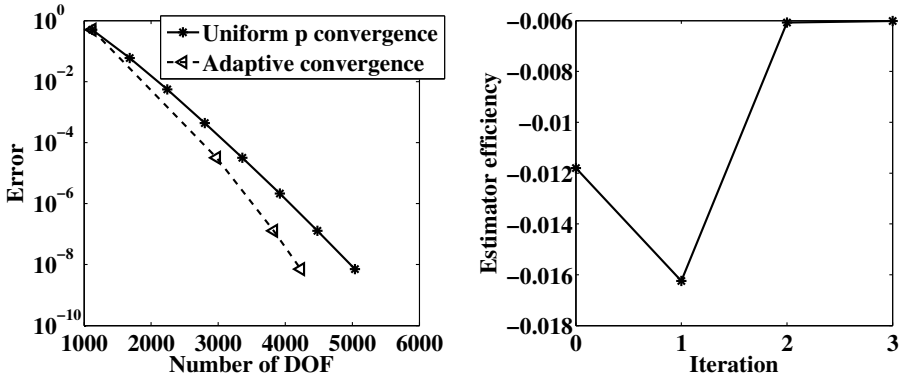


Figure 3.13: Wang flow problem: comparison of p-uniform computations, for $p = 1 \dots 8$, and a p-adaptive computation (left), and error estimator efficiency in the three iterations of the adaptive computation (right).

form $p = 1$ mesh, the desired accuracy is reached with three iterations of the adaptive process. In the left figure, the convergence to the exact solution obtained with the p-adaptive computation is compared with the convergence of the p-uniform computations, for $p = 1 \dots 8$. The maximum elemental error is plotted versus the number of coupled DOF, that is, the size of the linear system of the HDG discretization. Even in this simple example, the adaptive technique reduces the computational complexity respect to the p-uniform computation, involving a smaller number of DOF for a given accuracy. The right figure shows the efficiency of the estimator in each iteration,

which is always greater than 98%.

In Figure 3.14, the p-map is shown for each iteration. As expected, higher approximation degrees are concentrated to the bottom of the domain, to properly capture the sharp gradient in the solution. The estimated and exact elemental error maps are also shown for each iteration, revealing again an excellent performance of the estimator.

Estimation of the NACA 0012 aerodynamic characteristics

The second numerical test concerns the evaluation of the aerodynamic characteristics of the NACA 0012 airfoil, introduced in Section 2.3.2, in laminar flow regime. In this case, a mesh with 560 elements is used and a geometrical $p = 2$ representation is considered.

Given that the quantities of interest are the aerodynamic characteristics of the airfoil, the region of interest $\Omega_{\texttt{int}}$ is taken as an area surrounding the airfoil composed by 5 element layers, thus avoiding excessive iterations to refine the outer part of the domain. However, the adaptive process is applied to the whole mesh, leading to a satisfactory uniform error in the whole domain, and minimizing the pollution effects of the error in the wake.

Reference values of the aerodynamic coefficients are obtained using a mesh with element size $h/2$, and uniform $p = 10$. The first simulation is carried out at Re=5 000 with angle of attack $\alpha = 2°$. The adaptive simulation is started with uniform degree $p = 2$, setting the tolerance $\varepsilon = 10^{-4}$, uniform elemental tolerance $\epsilon_i = \varepsilon$ for $i = 1, \ldots, \texttt{n}_{\texttt{el}}$, the adaptation parameter $b = 10$ and the maximum degree $p^+ = 10$.

Convergence is achieved in 5 iterations. Figure 3.15 shows the solution and the p-map in different parts of the domain. As expected, higher order elements are automatically placed in the boundary layer and in the wake. Thus, the boundary layer at the leading edge and the recirculation bubbles at the trailing edge are correctly captured. Note that thanks to the high-order elements that are automatically placed just after the leading edge of the airfoil there is no need for highly stretched elements to resolve the boundary layer.

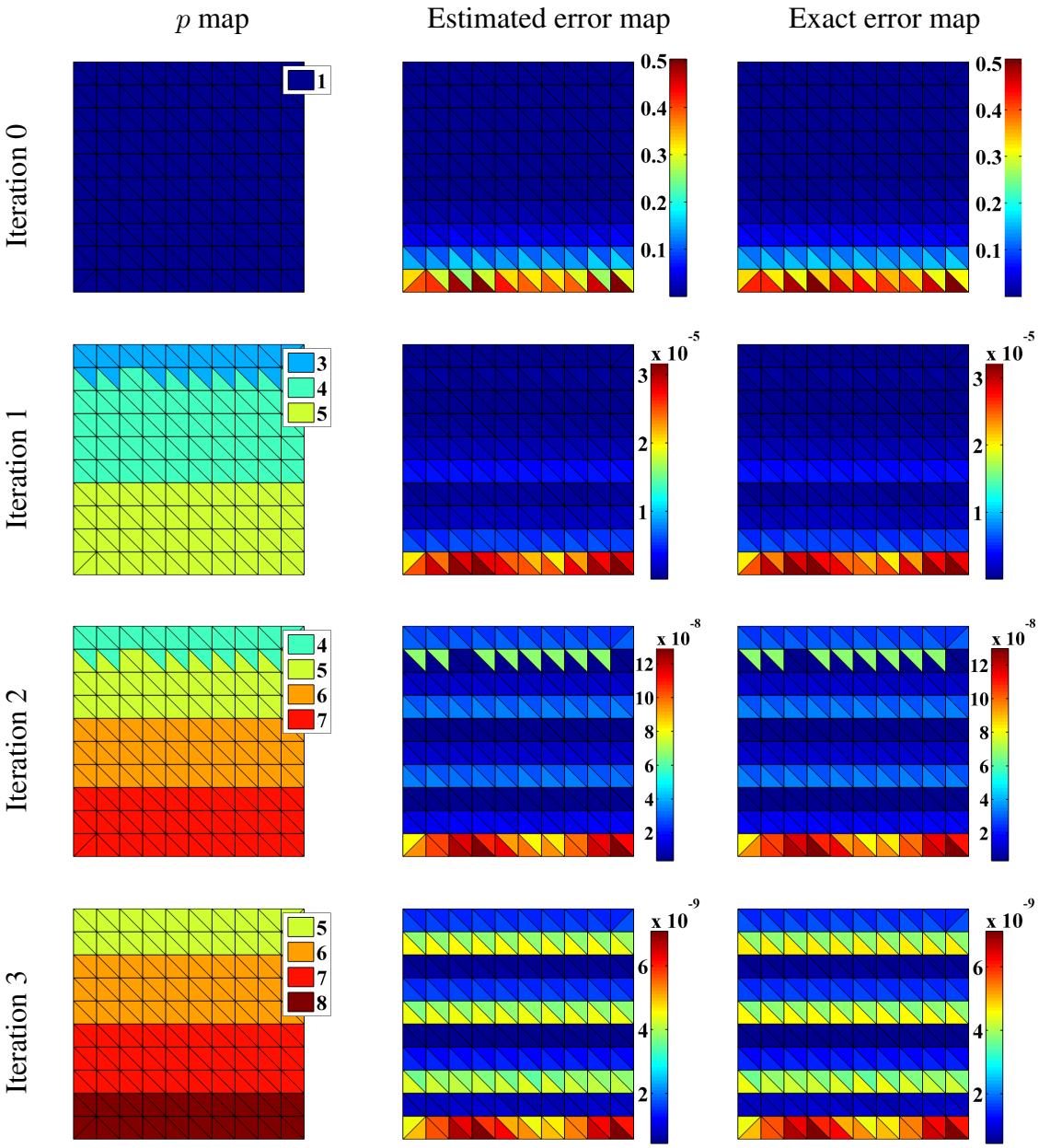


Figure 3.14: Wang flow problem: p-map, estimated error map and exact error map in each iteration of the adaptive simulation.

Table 3.2: NACA 0012 airfoil at Re=5 000 and $\alpha = 2°$: runtime for the solve and error estimation for each iteration.

iteration	0	1	2	3	4	5
runtime solve [s]	139	40	38	36	36	41
runtime err. est.[s]	3	5	5	5	5	6

Figure 3.16 shows the convergence to the reference values of the lift coefficient C_L and the drag coefficient C_D, for p-uniform computations with $p = 2 \dots 10$, and the p-adaptive computation. Fast convergence to the reference solution is obtained with the adaptive simulation, reaching an error smaller than 0.1% in the C_L and than 0.01% in the C_D. Similar accuracy is obtained with a uniform $p = 5$ computation, but with about 40% more DOF than the adaptive computation.

Thus, it is important to note that even though the error estimator is only based on the velocity error, the fluid dynamic force on the airfoil, that involves integrals of the pressure and the velocity gradient on the airfoil boundary, are properly captured.

To evaluate the computational overhead of computing the error with the proposed estimator, in Table 3.2 are shown the runtimes for the solution and for the error estimation for each iteration. As explained in Remark 7, for the second and subsequent iterations of the adaptive process the initial guess of the solution, $\boldsymbol{u}_0$, is the solution of the previous iteration. Thus, the initial guess is accurate enough to solve the stationary version of the Navier-Stokes equations, reducing the runtime for the solve. In any case, the computational cost of the error estimation is significantly smaller then the cost of computing the solution in all the iterations.

The second simulation has been carried out on the same computational mesh, with Re=10 000 and 2° angle of attack. The same parameters and tolerance are used to set up the adaptive process. Convergence is achieved in 5 iterations. A detail of the p-map is shown in Figure 3.17: note that respect to the results with Re=5 000 depicted in Figure 3.15, now higher order elements are automatically placed at the leading edge and at the trailing edge to capture the thinner boundary layer. The aerodynamic coefficients are shown in Figure 3.18. In this case, the adaptive simulation requires less than 70% DOF of the uniform $p = 6$ computation with similar accuracy. Figure

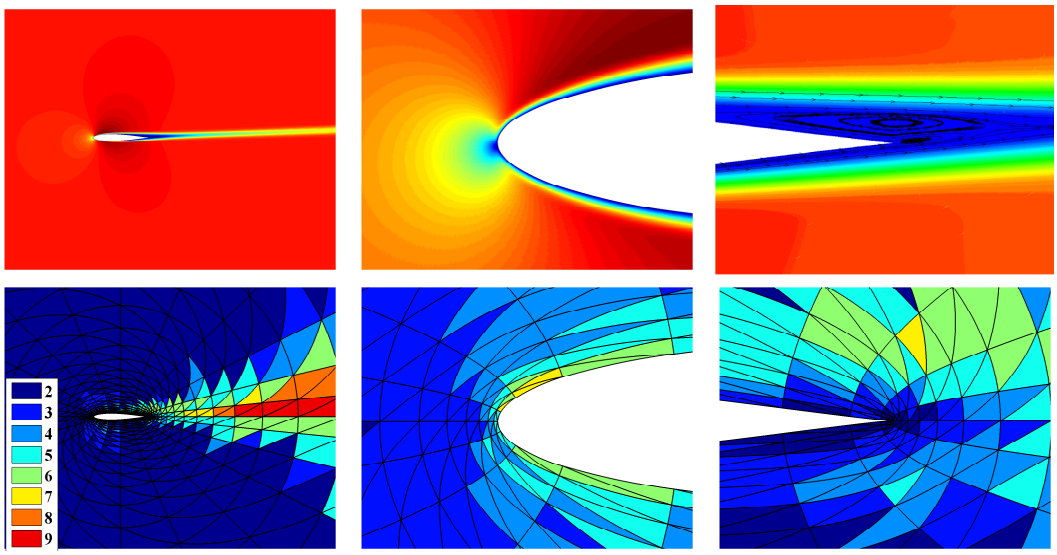


Figure 3.15: NACA 0012 airfoil: solution of the adaptive computation for the at Re=5 000 and $\alpha = 2°$, after convergence in 5 iterations with tolerance 10^{-4}. Velocity module (up) and p-map (down) around the airfoil and details around the leading edge (center) and the trailing edge (right).

3.19 shows the pressure and the skin friction coefficients around the airfoil, in very good agreement with the reference values.

To underline the importance of the refinement in the wake, the simulation at Re=10 000 is also performed reducing the elemental tolerance ϵ_i in the outer elements, that is, $\epsilon_i = 10^{-4}$ for $\Omega_i \subset \Omega_{\texttt{int}}$ and $\epsilon_i = 10^{-2}$ for $\Omega_i \subset \Omega \setminus \Omega_{\texttt{int}}$. No convergence is achieved in this case and the computation is stopped when in two successive iterations the degree remains unchanged for 99% of the elements. The resulting solution, see Figure 3.20, presents an important error compared with the reference solution. Even though globally the pressure coefficient seems to be well captured, the error in the pressure coefficient (which is shown in the zoom) produces an overestimate of the C_L of 50%.

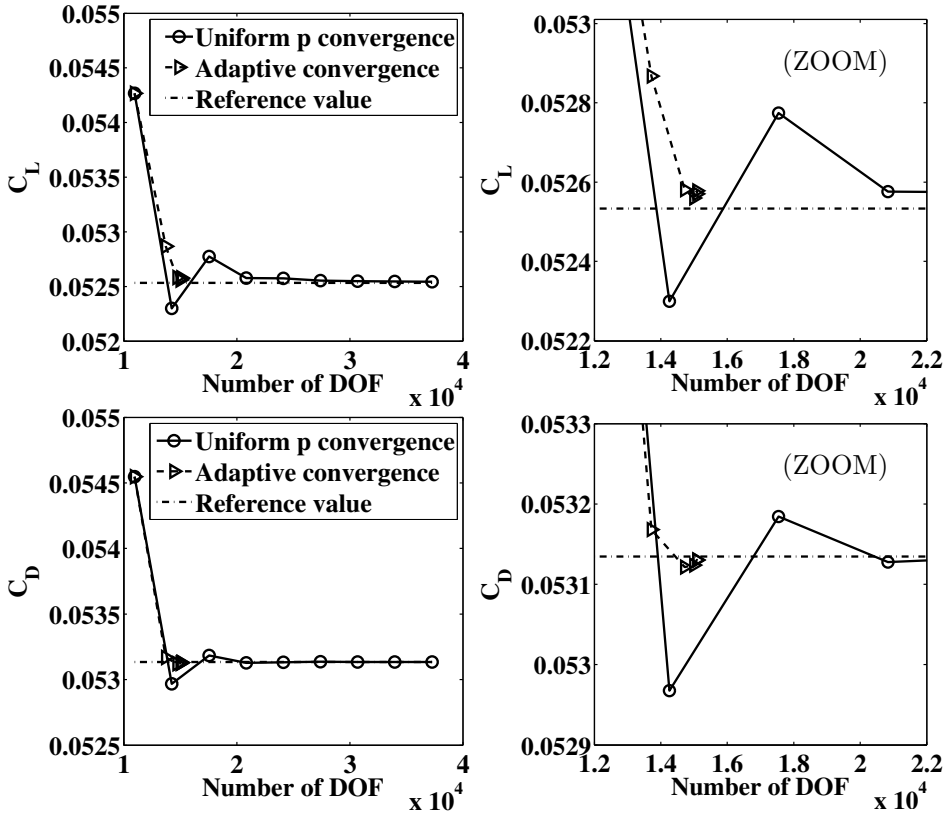


Figure 3.16: NACA 0012 airfoil: convergence of the aerodynamic coefficients C_L and C_D at Re=5 000 and $\alpha = 2°$. Non-adaptive and adaptive simulations are compared with a reference solution. The right figures depict a zoom of the left figures around the convergence point of the adaptive simulation.

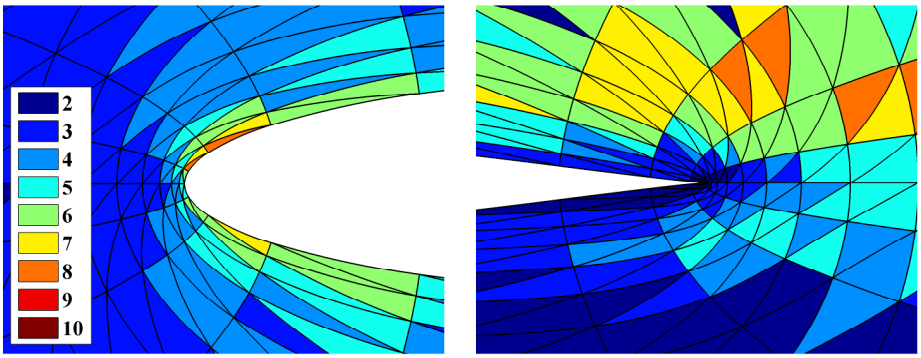


Figure 3.17: NACA 0012 airfoil at Re=10 000 and $\alpha = 2°$: map of p at the leading edge (left) and at the trailing edge (right).

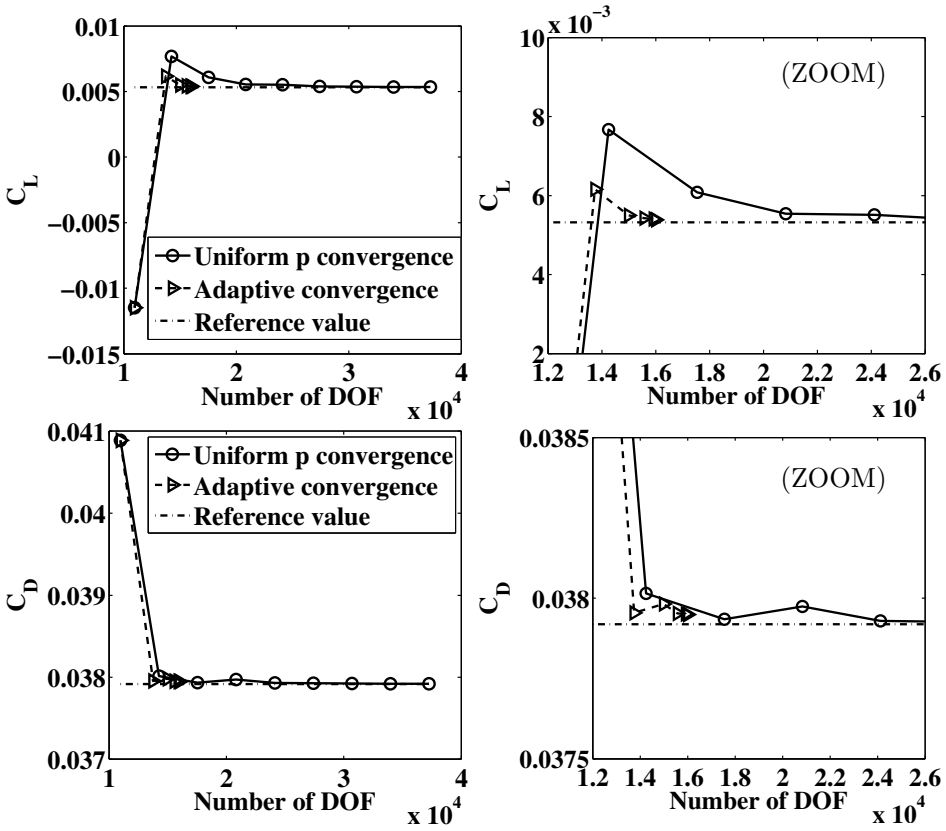


Figure 3.18: NACA 0012 airfoil at Re=10 000 and $\alpha = 2°$: convergence of the aerodynamic coefficients C_L and C_D. Non-adaptive and adaptive simulations are compared with a reference solution. The right figures depict a zoom of the left figures around the convergence point of the adaptive simulation.

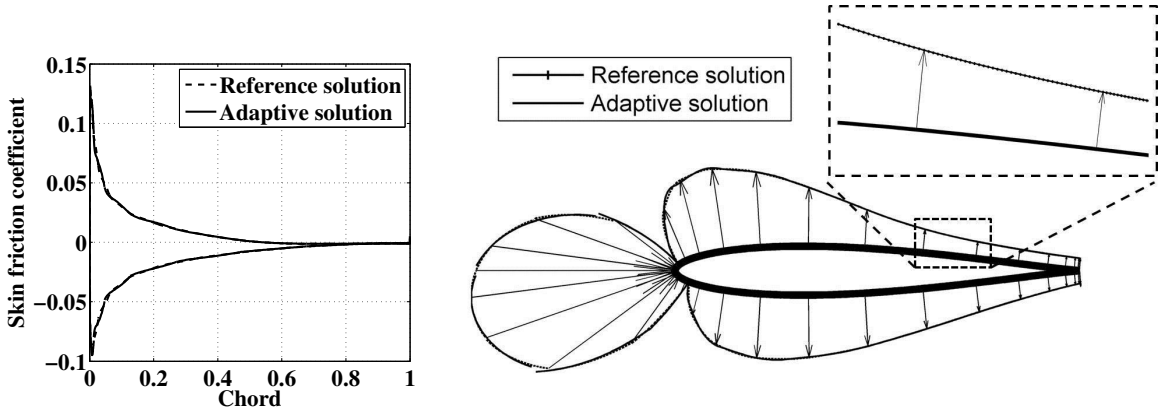


Figure 3.19: NACA 0012 airfoil at Re=10 000 and $\alpha = 2°$: skin friction coefficient (left) and pressure coefficient (right).

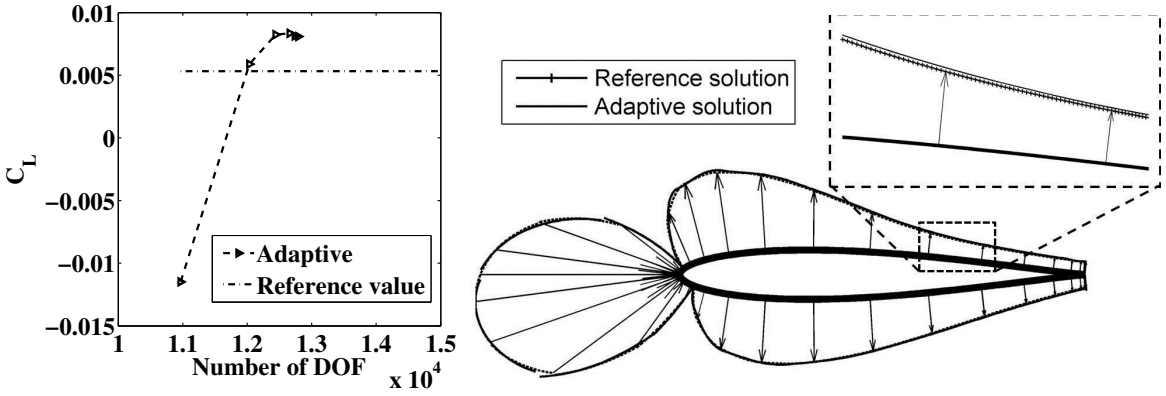


Figure 3.20: NACA 0012 airfoil at Re=10 000 and $\alpha = 2°$: non converging solution due to a too large error tolerance in the wake. C_L approximations (left) and final pressure coefficient (right).

Von Kármán street adaptive simulation

The objective of this example is to study the performance of the proposed p-adaptive technique in a transient simulation. The problem considered is the periodic vortex shedding of a flow past a unitary radius cylinder at Re=100. The computational mesh, illustrated in Figure 3.21, is composed by 728 triangular elements of variable element size, and it is refined around the cylinder which is represented by 12 elements of $p = 2$ geometry. Since space discretization errors are of concern here, no time step adaptation is considered, and a backward Euler method with fix time step $\Delta t = 10^{-2}$ is used. The simulation is started incrementing the Re number from 0.5, reaching the prescribed value in 200 time steps. The adaptive scheme is set up with tolerance $\varepsilon = 10^{-3}$, uniform elemental tolerance $\epsilon_i = \varepsilon$ for $i = 1, \ldots, \mathtt{n_{el}}$, adaptation parameter $b = 10$ and minimum $p^- = 2$. No upper bound for the degree is prescribed in this case. The adaptive refinement is performed every 10 time steps.

Results of the adaptive simulation are compared with four p-uniform simulations, with $p = 3, 4, 5, 6$, considering the uniform $p = 6$ solution as the reference solution. Figure 3.22 depicts the variation in time of the number of DOF of the adaptive computation, compared with the constant DOF of the p-uniform computations. Analyzing the evolution of number of DOF of the adaptive simulation it is possible to recognize

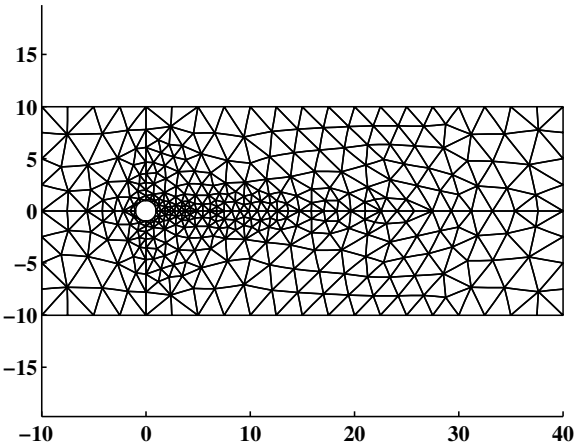


Figure 3.21: Von Kármán street example: computational mesh.

three phases: in the first phase the number of DOF increases smoothly and reaches a constant value. In this part of the simulation, the cylinder is not shedding and the adaptive technique is placing high-order elements around the cylinder and in the recirculation zone behind it. The rapid increase around time $t = 300$ is reflecting the start of the shedding pattern. After that, the number of DOF reaches a constant value again, indicating that the periodic state is reached. In other words, the complexity of the solution is reflected in the number of DOF that the adaptive technique automatically places in the domain, underlining once more the efficiency of this methodology for transient problems. Figure 3.23 shows the solution and the corresponding p-map in two time steps, before and after the start of the vortex shedding.

Table 3.3 shows the error in the mean drag force on the cylinder, averaged in one period of the periodic shedding regime. The adaptive simulation provides the best accuracy with almost the same number of DOF of the $p = 3$ uniform computation, that is 58% of the number of DOF of the $p = 6$ computation. On the other hand, the adaptive computation provides an accuracy similar to a $p = 6$ computation with almost half DOF.

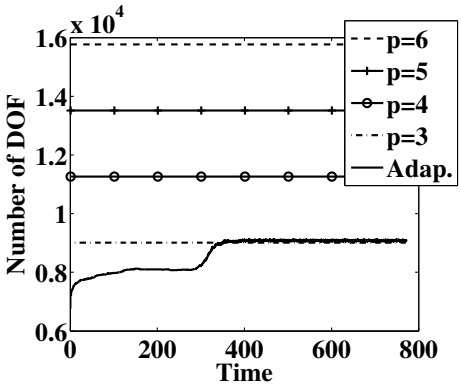


Figure 3.22: Von Kármán street past a cylinder at Re=100: evolution in time of the number of DOF in the adaptive and p-uniform simulations, for $p = 3, 4, 5, 6$.

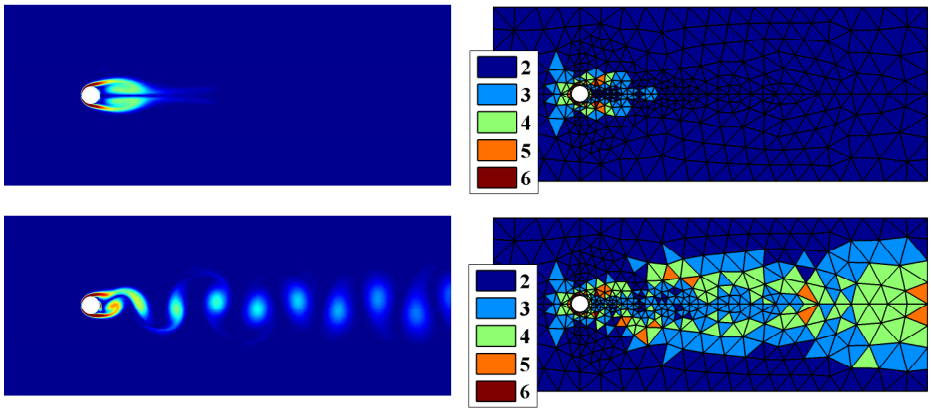


Figure 3.23: Von Kármán street past a cylinder at Re=100: vorticity pattern (left) and p-map (right) in a time step before the start of the shedding ($t = 14$ up Figures) and after it ($t = 750$ down Figures).

Table 3.3: Von Kármán street past a cylinder at Re=100: error in the mean value of the drag in one period of the vortex shedding flow, and % of number of DOF respect to the reference $p = 6$ solution.

	$p = 3$	$p = 4$	$p = 5$	Adaptive
Mean drag error %	1.8	0.52	0.13	0.014
DOF %	57	71	86	58

3.4 Conclusions

A HDG p-adaptive technique is proposed for the solution of harbor agitation problems, which require solving the Helmholtz equation with non-constant coefficients in an unbounded domain. The computational challenges of this problem are due to large domains, compared to the wavelength, and small geometrical features, inducing thus very large systems of equations. Moreover, the non-uniform wavelength requires different spatial resolutions in different areas, and complex geometries generates an unpredictable scattering pattern. The proposed adaptive technique provides uniform error distributions below a user defined tolerance, automatically placing the correct polynomial degree in each element. This allows to capture the oscillations accordingly with the wavelength and the complexity of the solution locally. Adaptivity is driven by an error estimator derived exploiting the superconvergent properties of HDG, and involving only elemental computations. Thus, the error estimation is an inexpensive computation compared to the solution. The efficiency of the proposed technique is superior to high-order continuous Galerkin, both from the point of view of the linear system dimension and the runtime. This is, to the authors knowledge, one of the first real engineering applications where discontinuous Galerkin is shown to outperform continuous Galerkin.

The HDG p-adaptive methodology is also applied to the solution of the Navier-Stokes equations. In this case, the complexity lies, on one hand, in the convective character of the equations that produces sharp localized gradients of the solution, on the other hand, the transient nature that requires to capture the complex structures of the flow as they move in the domain. The p-adaptive technique tackles both these difficulties: it relaxes the need of a priori design the discretization to correctly resolve the solution, for example with the use of highly stretched elements. Also, it allows to repeatedly update the discretization, without any topological change in the mesh, to follow the flow as it evolves. Three 2D numerical tests are shown. High efficiency of the error estimator is illustrated using an academic test with known analytical solution, for both uniform and p-variable meshes. The methodology is then applied to two

examples of engineering interest: the estimation of the aerodynamic characteristics of a NACA 0012 airfoil in laminar flow, and the evaluation of the drag force on a shedding cylinder. High accuracy and reduction in computational cost, compared with non-adaptive simulations, is found both for steady state and transient simulations. Numerical results also demonstrate that, even though the adaptive process is based on an error estimate only for the velocity field, accurate approximations of the fluid-dynamic forces are obtained.

Chapter 4

Summary and future developments

This thesis presents the development of an HDG p-adaptive technique for the solution of wave propagation problems and incompressible flow problems. The adaptive technique is based on the modification of the polynomial approximation in each element and face of the mesh, and it is driven by an error estimator, based on the superconvergent properties of HDG, that involves only elemental computation, thus it is computationally inexpensive compared to the solution itself. In the following, the main contributions of the thesis are summarized.

- **p-adaptive HDG for wave problems**: The Mild Slope equation, which derivation is detailed in Appendix F, is considered for modeling the wave propagation in infinite domains and inhomogeneous media. The MSE in time-frequency formulation is studied in Appendix E using a DG method with upwind fluxes. For the frequency formulation of the MSE, high-order CDG and HDG are derived. In particular, for HDG, the hybridization technique allows to reduce the global coupled unknowns to those of an approximation of the solution defined only on the edges of the mesh, reducing drastically the dimension of the linear system. Furthermore, the HDG solution is suitable for a local post-process resulting in a superconvergent solution. The accuracy of the solution and the post-process solution depends on a stabilization parameter, which influence in studied in Appendix A.

The computational efficiency of HDG is compared to the one of CDG and CG with static condensation of the interior nodes in Section 2.2. Numerical examples include an academic test with analytical solution and a geometry with corners. The latter induces an unbounded solution with the consequent loss of optimal convergence rate. The MSE also allows to solve problems of engineering interest such as sea wave propagation from the open sea to the interior of an harbor. The code has been developed in Matlab, and some details on the implementations are shown in Appendix B. In order to tackle real size applications of high computing requirements, a code optimization, detailed by Roca et al. (2011), has been implemented in the code. This optimization is based on the substitution of all the code loops (loop in elements, loop in faces, loop in Gauss points) with matrix-matrix multiplications. This has permitted a thorough comparison between high-order HDG and high-order CG for real applications, based on the total runtime to compute the solution.

Encouraged by the excellent performance of high-order elements in HDG, and thanks to the discontinuous character of the approximate solution, a p-adaptive strategy has been proposed for HDG in Chapter 3. The proposed adaptive technique provides uniform error distributions below a user defined tolerance, automatically modifying the polynomial degree in each element. It is driven by an error estimator derived exploiting the HDG superconvergent postprocess and involving only elemental computations. Thus the error estimation is inexpensive compared to the solution of the problem. The p-adaptive HDG for wave problems outperforms high-order CG, both from the point of view of the linear system size and the runtime.

- **p-adaptive HDG for fluid problems**: high-order HDG for the solution of the incompressible Navier-Stokes equation has been implemented in 2D and 3D. Some indications on the implementation are given in Appendix C.

  Two fractional step methods are derived for the time integration of the unsteady Navier-Stokes equations in Appendix D. The proposed approach allows to cir-

cumvent the difficulties related to the solution of the saddle point linear system produced by the HDG discretization of the Navier-Stokes equations. Convergence rate and efficiency of the proposed method is shown, outperforming other well established approaches.

The p-adaptive technique is also applied to the solution of fluid problems in Section 3.3, for steady-state and transient solutions. Various numerical experiments are presented, including academic examples and more challenging applications of engineering interest. High efficiency of the error estimator is found using an analytical example. Moreover, even though the adaptive technique is driven by an error estimator involving only errors in the velocity field, high accuracy is obtained in the estimation of the fluid-dynamic forces, and sharp resolution of the key features of the flow. In the next section, dedicated to the ongoing work, the adaptive technique is used to solve a challenging application involving moving reference frames, and also a 3D example.

4.1 Ongoing work

4.1.1 Application of the p-adaptive technique to problems with rotating reference frames

Adaptive computation are particularly suited for moving geometries and reference frames. In fact, in this situations no *a priori* adaptation of the computational mesh is possible, and the complex flow structures cover, in different time instants, all the mesh.

Solving the Navier-Stokes equations in a rotating frame requires taking into account the fictitious force rising in the non-inertial frame of reference. Details on this topics can be found in Li et al. (2002). Here, the HDG p-adaptive method has been used to simulate a vertical axes wind turbine (VAWT) in 2D. The turbine is made up of four NACA 0012 symmetric airfoils. A dimensionless problem is considered. An angular velocity $\omega = 1$ and an upstream velocity $V_0 = 1$ are set. The Reynolds

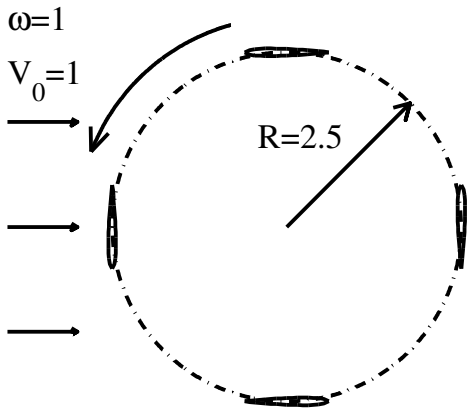


Figure 4.1: 2D VAWT: statement of the problem.

number is Re=1000. The statement of the problem is shown in Figure 4.1.

The simulation is carried out in a moving reference frame, which is rotating with the turbine. Thus, the body forces appearing in the computational domain generate a rotating velocity whose magnitude increases with the distance from the rotation center. For this reason, particular care must be taken with the choice of the stabilization parameter τ, see Section 2.3.1. In fact, numerical evidence shows that if a constant τ is used, either loss of stability or loss of superconvergence may deteriorate the numerical solution. Hence, a variable τ is used in this case, with the following relation

$$\tau = \begin{cases} \tau_0 \omega |\boldsymbol{x}| & \text{if } \omega|\boldsymbol{x}| \geq 1, \\ \tau_0 & \text{if } \omega|\boldsymbol{x}| < 1, \end{cases}$$

where $\boldsymbol{x}$ is the position vector pointing at the barycenter of the face on which τ is defined, and τ_0 is a constant value for the whole mesh.

The adaptive procedure is set up using a parameter $b = 10$, a tolerance $\varepsilon = 10^{-4}$ and a uniform elemental tolerance $\epsilon_i = \varepsilon$ for $i = 1 \cdots \mathtt{n_{el}}$. A constant time step $\Delta t = 10^{-2}$ is used, and the error estimation and adaptive procedure is performed every 10 time steps. The maximum and minimum values of the polynomial degree are $p^- = 1$ and $p^+ = 9$.

In Figure 4.2 is shown the vorticity and the p-map in an area of the domain

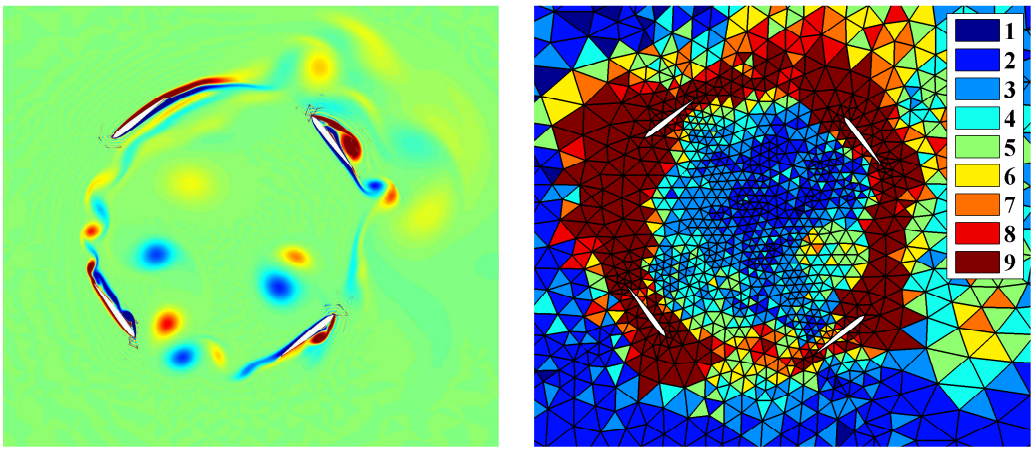


Figure 4.2: 2D VAWT: vorticity (left) and p-map (right) for $t = 4.3$.

surrounding the turbine blades, at time $t = 4.3$. The adaptive technique is correctly placing high-order elements around the blades and in the wake that they generate. However, for this example the mesh seems to be too coarse as the maximum p is reached in a large part of the domain.

In Figure 4.3 are shown the aerodynamic coefficient of the turbine during the simulation while Figure 4.4 shows the evolution of the number of DOF. As expected, after an initial transient, the aerodynamic coefficients reach an quasi oscillatory behavior and the number of DOF reaches an almost constant value. Figure 4.5 shows a frequency analysis of the lift and drag coefficients of the turbine. Even with this coarse mesh, a good agreement with reference solutions computed with the commercial code FLUENT[1] is found.

4.1.2 Application to 3D problems

The HDG p-adaptive technique has been extended to 3D problems. Some preliminary results are shown here for a 3D version of the lid-driven cavity test case. The lid-driven cavity is a standard benchmark for incompressible flows, see for example Montlaur et al. (2010). A cubic domain $\Omega =]0, 1[\times]0, 1[\times]0, 1[$ is considered, with zero body

[1]FLUENT data courtesy of Adeline de Villardi de Montlaur

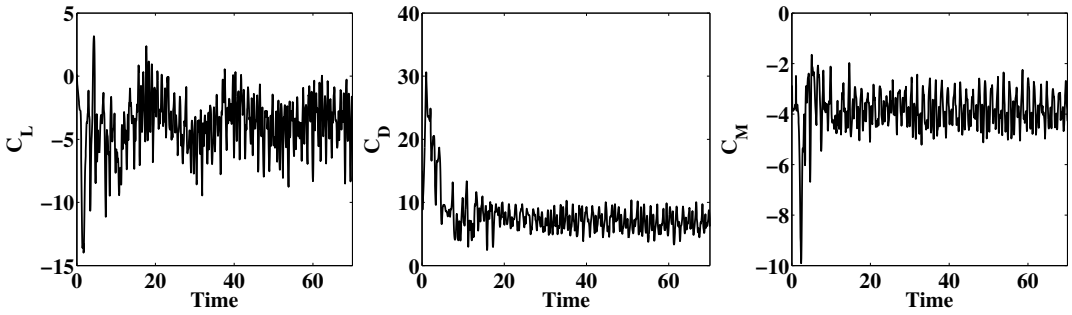


Figure 4.3: 2D VAWT: lift, drag and moment coefficients as a function of the time.

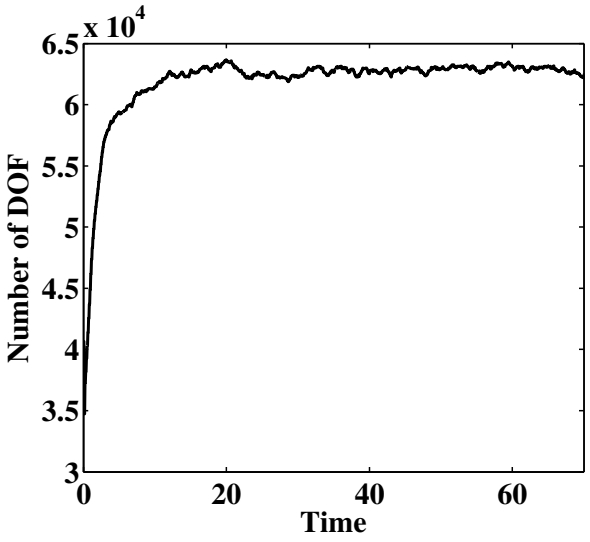


Figure 4.4: 2D VAWT: number of DOF as a function of the time.

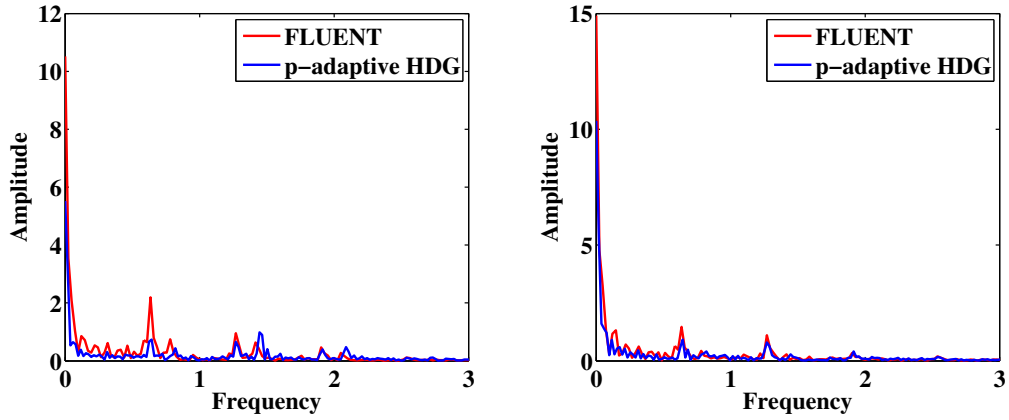


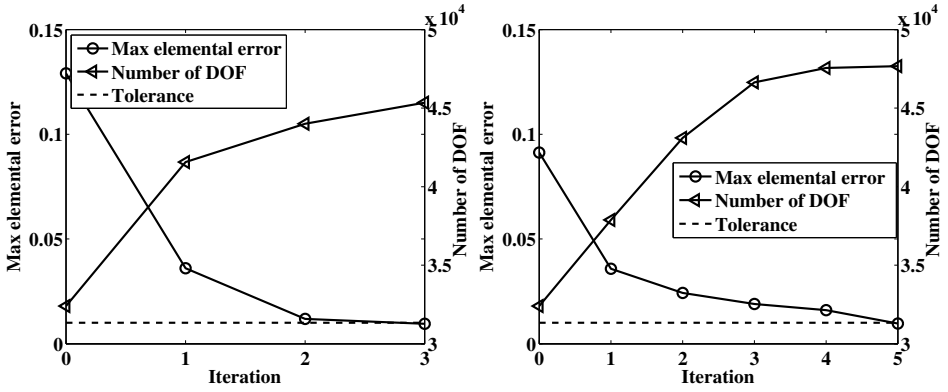
Figure 4.5: 2D VAWT: frequency analysis of the lift (left) and drag (right) coefficients and comparison with FLUENT results.

forces and one moving wall. A continuous velocity is imposed on the upper wall $z = 1$

$$\boldsymbol{u} = \begin{cases} \Theta(10x, 0, 0)^T & \text{for } 0 \leq x \leq 0.1, \\ \Theta(1, 0, 0)^T & \text{for } 0.1 \leq x \leq 0.9, \\ \Theta(10 - 10x, 0, 0)^T & \text{for } 0.9 \leq x \leq 1, \end{cases} \text{ with } \Theta = \begin{cases} 10y & \text{for } 0 \leq y \leq 0.1, \\ 1 & \text{for } 0.1 \leq y \leq 0.9, \\ 10 - 10y & \text{for } 0.9 \leq y \leq 1, \end{cases}$$

and a zero velocity $\boldsymbol{u} = (0, 0, 0)^T$ is enforced on the other walls. The computational mesh is made up of $10 \times 10 \times 10$ hexahedra. The adaptive simulation has been carried out setting the tolerance $\varepsilon = 10^{-2}$ and a constant elemental tolerance $\epsilon_i = \varepsilon$ for $i = 1 \cdots \mathtt{n_{el}}$. An adaptation aggressiveness $b = 100$ is set.

Figure 4.6 depicts the evolution of the maximum elemental error and the number of DOF during the iterations of the adaptive technique, for Re= 10 and Re=1000. As expected, the higher Re yields a higher number of iteration to achieve convergence of the adaptive technique (3 iterations for Re= 10, 5 iterations for Re= 1000) and also an larger number of DOF at convergence.

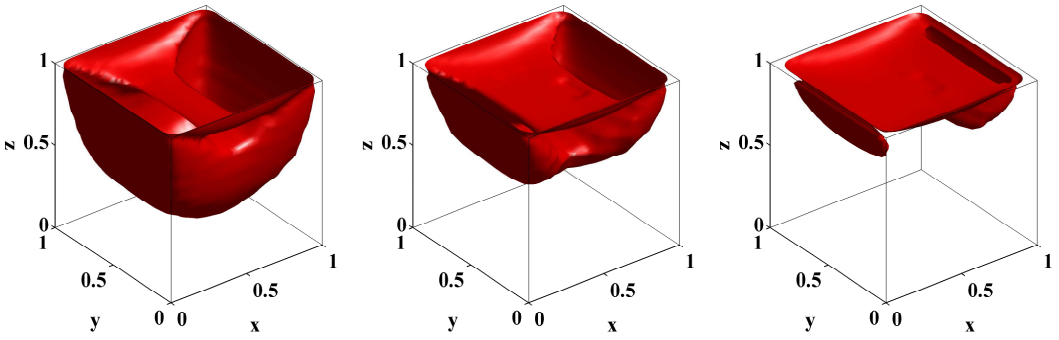


(a) $Re = 10$ (b) $Re = 1000$

Figure 4.6: 3D cavity flow: evolution of the maximum elemental error and the number of DOF in the p-adaptive simulations at Re= 10 and Re= 1000.

Figures 4.7 and 4.8 and Figures 4.9 and 4.10 show the iso-surfaces of the velocity magnitude at values $0.1, 0.2$ and 0.3, and the distribution of the polynomial degrees in the computational mesh, respectively for Re= 10 and Re= 1000. The higher velocity

propagates more deeply into the cavity increasing the Re, as expected. On the other hand, the adaptive technique places higher-order elements on the top of the cube, hence close to the moving wall, and at the edges where the velocity is subject to the strongest turns. In both cases, a symmetric distribution of the polynomial degrees with respect to the velocity direction is obtained.



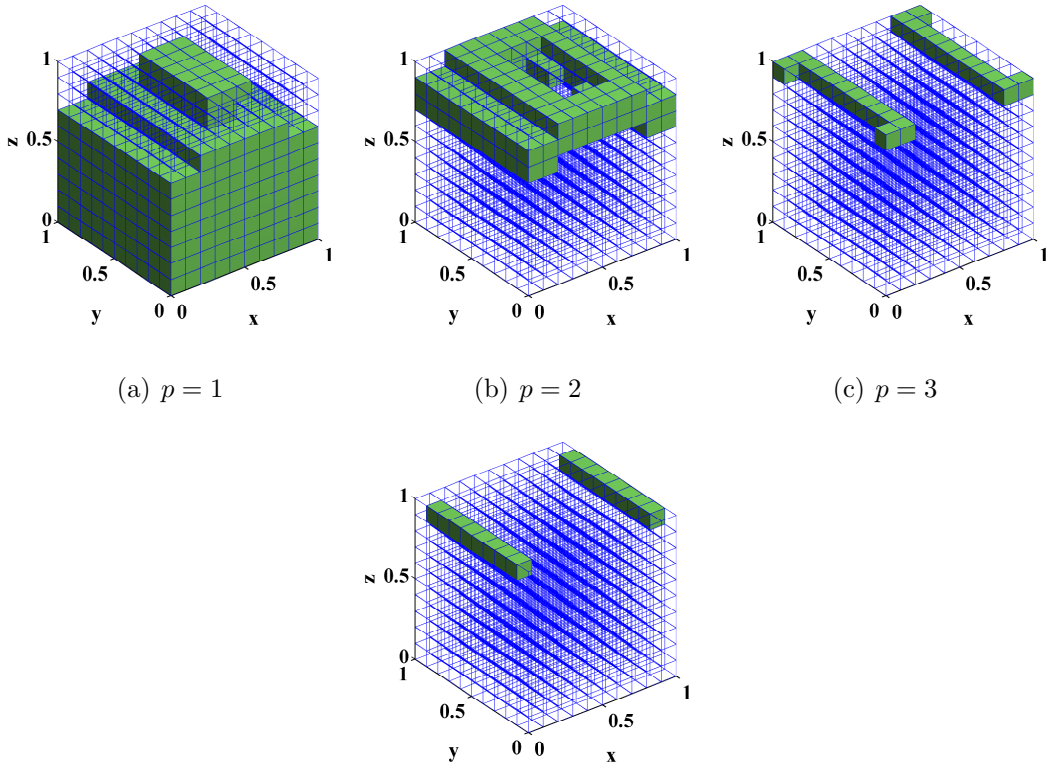
(a) $|\boldsymbol{u}| = 0.1$ (b) $|\boldsymbol{u}| = 0.2$ (c) $|\boldsymbol{u}| = 0.3$

Figure 4.7: 3D cavity flow Re= 10: iso-surfaces of velocity magnitude

4.2 Future developments

- **Goal oriented adaptivity**: recent developments in the field of a posteriori error estimation regards the use of duality arguments to obtain goal oriented error estimations. This approach is based on defining an output of interest of the numerical computation, normally defined by a functional of the solution. Goal-oriented error assessment strategies aim at estimating the error committed in these quantity of interest and possibly providing bounds for it.

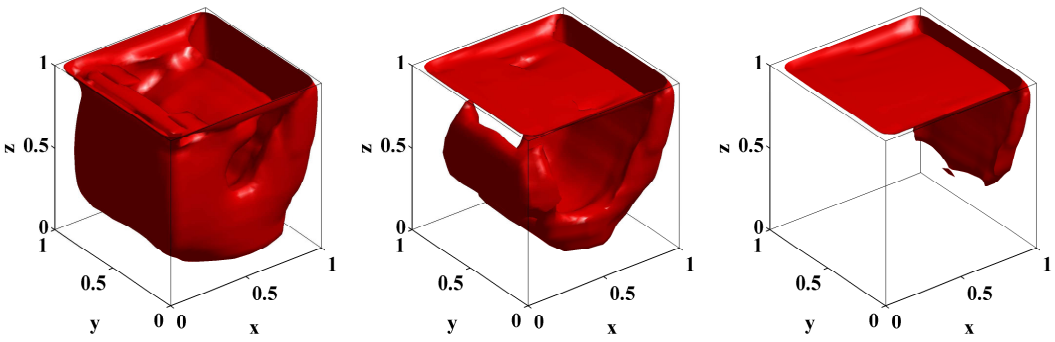
  The error propagates inside the area of interest from zone of the domain outside it, in particular for wave problems. The convenience of a goal oriented approach is well known in this context, see for example Steffens et al. (2011). However, the dual problem approach can effectively be used also for fluid problems. For example, for the NACA numerical test of Section 3.3, a non-linear functional



(a) $p = 1$ (b) $p = 2$ (c) $p = 3$

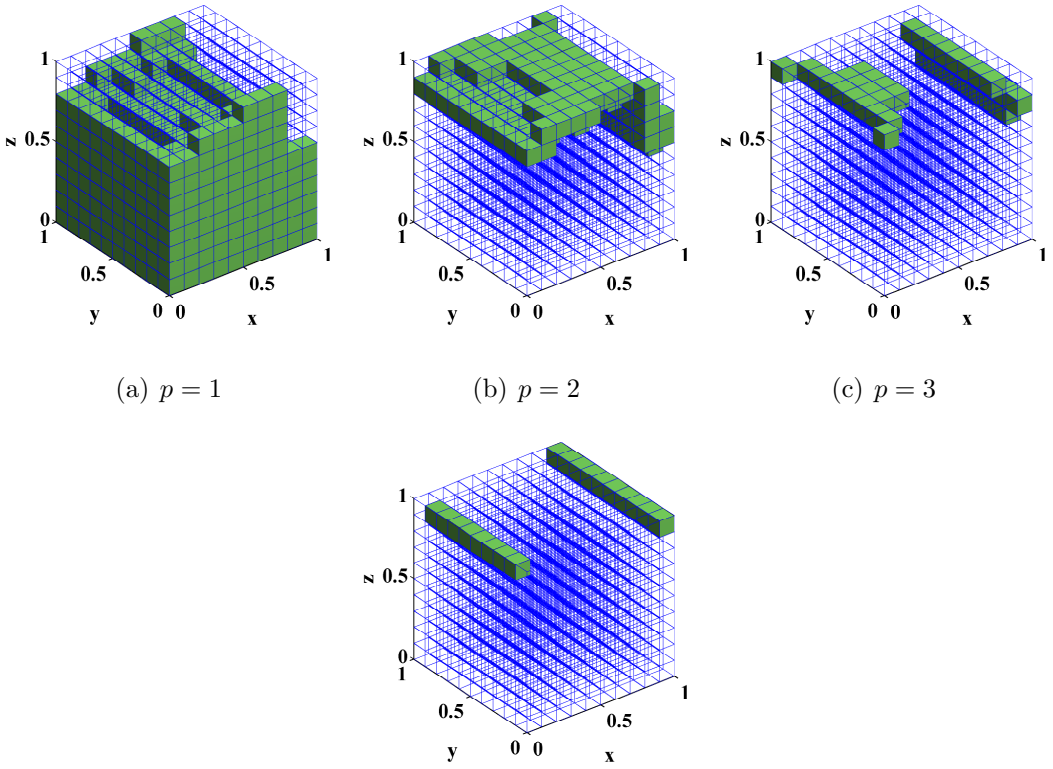
(d) $p = 4$

Figure 4.8: 3D cavity flow Re= 10: p-map in the last iteration of the adaptive procedure.



(a) $|\boldsymbol{u}| = 0.1$ (b) $|\boldsymbol{u}| = 0.2$ (c) $|\boldsymbol{u}| = 0.3$

Figure 4.9: 3D cavity flow Re= 1000: iso-surfaces of velocity magnitude.



(a) $p = 1$ (b) $p = 2$ (c) $p = 3$

(d) $p = 4$

Figure 4.10: 3D cavity flow Re= 1000: p-map in the last iteration of the adaptive procedure.

can be defined for each one of the aerodynamic characteristic, see for example Hartmann et al. (2011). A goal-oriented adaptive technique based on the solution of dual problem would allow to improve the performance of the adaptive strategy and reduce the need of user defined parameters.

- ***hp*-adaptivity**: p-adaptivity is very efficient for problems with smooth solution such as the ones considered in this thesis. To further improve the performance of the adaptive technique, a hp-adaptive technique could be implemented. This would reduce even more the requirements on the initial geometric discretization, since h-refinement would be used to reduce the element size in those elements where the maximum allowed polynomial degree is already used.

- **Parallelization**: the Matlab code optimization has been pushed to the limit where the performance are similar to a Fortran or C++ code *without parallelization*. This has allowed to solve large 2D problems and relatively small 3D problems. To be able to tackle large 3D problems, a complete migration to Fortran of C++ is required. The LaCàN C++/Matlab library will help in the migration process.

- **Extension to turbulent compressible flows**: nowadays, DG methods for compressible flows are a cutting-edge topic in research for aerospace applications. Adaptivity in this field is fundamental to be able to correctly capture the characteristic features of the flow such as shocks and boundary layers, see for example Hartmann (2006) and Wang and Mavriplis (2009). The application of HDG in this field is promising, see for example Peraire et al. (2010) and Nguyen and Peraire (2011a). Hence, the model problem should be modified accounting for turbulence, using the RANS equations for example (see Nguyen and Peraire (2011b)), and for compressibility, modifying the continuity equation and adding the energy equation.

Bibliography

Abarbanel, S. and D. Gottlieb (1998). On the construction of absorbing layers in CEM. *Appl. Numer. Math. 27*(4), 331–340.

Ainsworth, M. and J. T. Oden (2000). *A posteriori error estimation in finite element analysis*. Pure and Applied Mathematics (New York). New York: Wiley-Interscience [John Wiley & Sons].

Alvarez, G. B., A. F. D. Loula, E. G. D. do Carmo, and F. A. Rochinha (2006). A discontinuous finite element formulation for Helmholtz equation. *Comput. Methods Appl. Mech. Eng. 195*(33-36), 4018–4035.

Arnold, D. N. (1982). An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal. 19*(4), 742–760.

Arnold, D. N. and F. Brezzi (1985). Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates. *RAIRO Modél. Math. Anal. Numér. 19*(1), 7–32.

Babuška, I. and W. C. Rheinboldt (1978). Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal. 15*(4), 736–754.

Babuška, I. and W. C. Rheinboldt (1979). Adaptive approaches and reliability estimations in finite element analysis. *Comput. Methods Appl. Mech. Eng. 17–18*(0), 519–540.

Babuška, I. and S. A. Sauter (2000). Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM Rev. 42*(3), 451–484.

Babuška, I. and B. Szabo (1991). *Finite element analysis*. New York: John Wiley & Sons.

Bassi, F. and S. Rebay (1997). A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys. 131*(2), 267–279.

Berenger, J.-P. (1994). A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys. 114*(2), 185–200.

Berkhoff, J. C. W. (1972). Computation of combined refraction-diffraction. In *Proceedings of the 13th Coastal Engineering Conference*, Volume 1, ASCE: Vancouver, Canada, pp. 471–490.

Berkhoff, J. C. W. (1976). *Mathematical models for simple harmonic linear water waves: wave diffraction and refraction.* Ph. D. thesis, Delft Hydraulics Laboratory, Delft University of Technology, Delft, Netherlands.

Bermúdez, A., L. Hervella-Nieto, A. Prieto, and R. Rodríguez (2004). An exact bounded PML for the Helmholtz equation. *C. R. Math. 339*(11), 803–808.

Bettess, P. and O. C. Zienkiewicz (1977). Diffraction and refraction of surface waves using finite and infinite elements. *Int. J. Numer. Methods Eng. 11*(8), 1271–1290.

Bey, K. S. (1994). *An hp-adaptive discontinuous Galerkin method for hyperbolic conservation laws.* Ph. D. thesis, The University of Texas, Austin, Texas.

Bey, K. S. and J. T. Oden (1996). hp-version discontinuous Galerkin methods for hyperbolic conservation laws. *Comput. Methods Appl. Mech. Eng. 133*(3–4), 259–286.

Bokaris, J. and K. Anastasiou (2003). Solution of the hyperbolic mild-slope equation using the finite volume method. *Int. J. Numer. Methods Fluids 41*(3), 225–250.

Booij, N. (1983). A note on the accuracy of the mild-slope equation. *Coast. Eng. 7*(3), 191–203.

Bouillard, P. and F. Ihlenburg (1999). Error estimation and adaptivity for the finite element method in acoustics: 2d and 3d applications. *Comput. Methods Appl. Mech. Eng. 176*(1–4), 147–163.

Bowman, J. J., T. B. Senior, and P. L. E. Uslenghi (1987). *Electromagnetic and Acoustic Scattering by Simple Shapes.* New York: Hemisphere.

Burbeau, A. and P. Sagaut (2005). A dynamic p-adaptive discontinuous Galerkin method for viscous flow with shocks. *Comput. Fluids 34*(4-5), 401–417.

Cantwell, C. D., S. J. Sherwin, R. M. Kirby, and P. H. J. Kelly (2011). From h to p efficiently: selecting the optimal spectral/hp discretisation in three dimensions. *Math. Model. Nat. Phenom. 6*(3), 84–96.

Castillo, P., B. Cockburn, I. Perugia, and D. Schötzau (2000). An a priori error analysis of the local discontinuous Galerkin method for elliptic problems. *SIAM J. Numer. Anal 38*(5), 1676–1706.

Cessenat, O. and B. Despres (1998). Application of an ultra weak variational formulation of elliptic PDEs to the two-dimensional Helmholtz problem. *SIAM J. Numer. Anal. 35*(1), 255–299.

Chang, C. (1990). A least-squares finite element method for the Helmholtz equation. *Comput. Methods Appl. Mech. Eng. 83*(1), 1–7.

Chen, Y. and B. Cockburn (2012a). Analysis of variable-degree HDG methods for convection-diffusion equations. part I: Semimatching nonconforming meshes. *IMA J. Numer. Anal.*. to appear.

Chen, Y. and B. Cockburn (2012b). Analysis of variable-degree HDG methods for convection-diffusion equations. part II: Semimatching nonconforming meshes. *Math. Comp.*. to appear.

Chorin, A. J. (1968). Numerical solution of the Navier-Stokes equations. *Math. Comp. 22*, 745–762.

Cockburn, B., B. Dong, and J. Guzmán (2008). A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. *Math. Comp. 77*(264), 1887–1916.

Cockburn, B., B. Dong, and J. Guzmán (2009). A hybridizable and superconvergent discontinuous Galerkin method for biharmonic problems. *J. Sci. Comput. 40*(1-3), 141–187.

Cockburn, B. and J. Gopalakrishnan (2005). A characterization of hybridized mixed methods for second order elliptic problems. *SIAM J. Numer. Anal. 42*(1), 283–301.

Cockburn, B., J. Gopalakrishnan, and R. Lazarov (2009). Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J. Numer. Anal. 47*(2), 1319–1365.

Cockburn, B., J. Guzmán, and H. Wang (2009). Superconvergent discontinuous Galerkin methods for second-order elliptic problems. *Math. Comput. 78*, 1–24.

Cockburn, B., G. Kanschat, I. Perugia, and D. Schotzau (2002). Superconvergence of the local discontinuous Galerkin method for elliptic problems on cartesian grids. *SIAM J. Numer. Anal. 39*(1), 264–285.

Cockburn, B. and C. W. Shu (1998). The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal. 35*(6), 2440–2463.

Codina, R. (2001). Pressure stability in fractional step finite element methods for incompressible flows. *J. Comput. Phys. 170*(1), 112–140.

Costabel, M. and M. Dauge (2000). Singularities of electromagnetic fields in polyhedral domains. *Arch. Ration. Mech. Anal. 151*, 221–276.

Díez, P., M. Arroyo, and A. Huerta (2000). Adaptivity based on error estimation for viscoplastic softening materials. *Mech. Cohesive-Fricct. Mater. 5*(2), 87–112.

Díez, P., J. J. Rodenas, and O. C. Zienkiewicz (2007). Equilibrated patch recovery error estimates: simple and accurate upper bounds of the error. *Int. J. Numer. Methods Eng. 69*(10), 2075–2098.

Dingemans, M. W. (1997). *Water wave propagation over uneven bottoms*, Volume 13 of *Advanced Series on Ocean Engineering*. World Scientific.

Donea, J. and A. Huerta (2003). *Finite element methods for flow problems*. Chichester (UK): John Wiley & Sons.

Dunavant, D. A. (1985). High degree efficient symmetrical Gaussian quadrature rules for the triangle. *Int. J. Numer. Methods Eng. 21*(6), 1129–1148.

Eskilsson, C. (2011). An hp-adaptive discontinuous Galerkin method for shallow water flows. *Int. J. Numer. Methods Fluids 67*(11), 1605–1623.

Farhat, C., I. Harari, and U. Hetmaniuk (2003). A discontinuous Galerkin method with Lagrange multipliers for the solution of Helmholtz problems in the mid-frequency regime. *Comput. Methods Appl. Mech. Eng. 192*(11-12), 1389–1419.

Feng, X. and H. Wu (2009). Discontinuous Galerkin methods for the Helmholtz equation with large wave number. *SIAM J. Sci. Comput. 47*(4), 2872–2896.

Fidkowski, K. J. and D. L. Darmofal (2007). A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations. *J. Comput. Phys. 225*(2), 1653–1672.

Fischer, P. F., G. W. Kruse, and F. Loth (2002). Spectral element methods for transitional flows in complex geometries. *J. Sci. Comput. 17*, 81–98.

Gabard, G. (2007). Discontinuous Galerkin methods with plane waves for time-harmonic problems. *J. Comput. Phys. 225*(2), 1961–1984.

Gabard, G., P. Gamallo, and T. Huttunen (2011). A comparison of wave-based discontinuous Galerkin, ultra-weak and least-square methods for wave problems. *Int. J. Numer. Methods Eng. 85*(3), 380–402.

Gago, J. P. d. S. R., D. W. Kelly, O. C. Zienkiewicz, and I. Babuška (1983). A posteriori error analysis and adaptive processes in the finite element method: Part II-adaptive mesh refinement. *Int. J. Numer. Methods Eng. 19*(11), 1621–1656.

Giorgiani, G., S. Fernández-Méndez, and A. Huerta (2012). Hybridizable discontinuous Galerkin p-adaptivity for wave problems. In *Proceedings of the ECCOMAS Young Investigators Conferences 2012*, Aveiro, Portugal.

Giorgiani, G., S. Fernández-Méndez, and A. Huerta (2013a). Hybridizable discontinuous Galerkin p-adaptivity for the incompressible Navier-Stokes equations. submitted.

Giorgiani, G., S. Fernández-Méndez, and A. Huerta (2013b). Hybridizable discontinuous Galerkin p-adaptivity for wave propagation problems. *Int. J. Numer. Methods Fluids*. Accepted for publication.

Giorgiani, G., D. Modesto, S. Fernández-Méndez, and A. Huerta (2013). High-order continuous and discontinuous Galerkin methods for wave problems. submitted.

Givoli, D. (1992). *Numerical methods for problems in infinite domains*, Volume 33 of *Studies in Applied Mechanics*. Amsterdam: Elsevier Scientific Publishing Co.

Givoli, D. and B. Neta (2003). High-order non-reflecting boundary scheme for time-dependent waves. *J. Comput. Phys. 186*(1), 24–46.

Gratsch, T. and K.-J. Bathe (2005). A posteriori error estimation techniques in practical finite element analysis. *Comput. Struct. 83*(4-5), 235–265.

Guermond, J. L., P. Minev, and J. Shen (2004). An overview of projection methods for incompressible flows. *Comput. Methods Appl. Mech. Engrg 41*, 112–134.

Harari, I. and K. Gosteev (2007). Bubble-based stabilization for the Helmholtz equation. *Internat. J. Numer. Methods Engrg. 70*(10), 1241–1260.

Harari, I. and T. J. R. Hughes (1992). Galerkin/least-squares finite element methods for the reduced wave equation with nonreflecting boundary conditions in unbounded domains. *Comput. Methods Appl. Mech. Eng. 98*(3), 411–454.

Harari, I. and F. Magoulès (2004). Numerical investigations of stabilized finite element computations for acoustics. *Wave Motion 39*(4), 339–349.

Hartmann, R. (2006). Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations. *Int. J. Numer. Methods Fluids 51*(9-10), 1131–1156.

Hartmann, R., J. Held, and T. Leicht (2011). Adjoint-based error estimation and adaptive mesh refinement for the RANS and $k-\omega$ turbulence model equations. *J. Comput. Phys. 230*(11), 4268–4284.

Hartmann, R., J. Held, T. Leicht, and F. Prill (2010). Discontinuous Galerkin methods for computational aerodynamics – 3D adaptive flow simulation with the DLR PADGE code. *Aerosp. Sci. Technol. 14*(7), 512–519.

Hartmann, R. and P. Houston (2002a). Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comput. 24*(3), 979–1004.

Hartmann, R. and P. Houston (2002b). Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys. 183*(2), 508–532.

Hesthaven, J. and T. Warburton (2002). Nodal high-order methods on unstructured grids: I. time-domain solution of Maxwell's equations. *J. Comput. Phys. 181*(1), 186–221.

Higham, N. J. (2002). *Accuracy and stability of numerical algorithms* (Second ed.). Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

Houston, P. and E. Süli (2001). hp-adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems. *SIAM J. Sci. Comput. 23*(4), 1226–1252.

Huerta, A., X. Roca, A. Aleksandar, and J. Peraire (2012). Are High-order and Hybridizable Discontinuous Galerkin methods competitive? *Oberwolfach Rep. 9*(1), 485–487. Abstracts from the workshop held February 12–18, 2012, Organized by Olivier Allix, Carsten Carstensen, Jörg Schröder and Peter Wriggers, Oberwolfach Reports. Vol. 9, no. 1.

Huerta, A., A. Rodriguez-Ferran, P. Díez, and J. Sarrate (1999). Adaptive finite element strategies based on error assessment. *Int. J. Numer. Methods Eng. 46*(10), 1803–1818.

Ihlenburg, F. and I. Babuška (1995a). Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation. *Int. J. Numer. Methods Eng. 38*(22), 3745–3774.

Ihlenburg, F. and I. Babuška (1995b). Finite element solution of the Helmholtz equation with high wave number Part I: The h-version of the FEM. *Comput. Math. Appl. 30*(9), 9–37.

Ihlenburg, F. and I. Babuška (1997). Finite element solution of the Helmholtz equation with high wave number Part II: The h-p version of the FEM. *SIAM J. Numer. Anal. 34*(1), 315–358.

Khellaf, M. C. and M. Bouhadef (2004). Modified mild slope equation and open boundary conditions. *Ocean Eng. 31*(13), 1713–1723.

Kim, K., D. Yi, and S. Lee (2005). Mortar method for nonconforming finite elements. *Appl. Math. Comput. 167*(1), 650–669.

Kirby, R., S. J. Sherwin, and B. Cockburn (2011). To CG or to HDG: A comparative study. *J. Sci. Comput. 51*(1), 183–212.

Kubatko, E. J., S. Bunya, C. Dawson, and J. J. Westerink (2009). Dynamic p-adaptive Runge-Kutta discontinuous Galerkin methods for the shallow water equations. *Comput. Methods Appl. Mech. Eng. 198*(21-26), 1766–1774.

Kubatko, E. J., S. Bunya, C. Dawson, J. J. Westerink, and C. Mirabito (2009). A performance comparison of continuous and discontinuous finite element shallow water models. *J. Sci. Comput. 40*(1-3), 315–339.

Kucukcoban, S. and L. Kallivokas (2011). Mixed perfectly-matched-layers for direct transient analysis in 2D elastic heterogeneous media. *Comput. Meth. Appl. Mech. Eng. 200*(1-4), 57–76.

Lacasse, D., A. Garon, and D. Pelletier (2007). Development of an adaptive discontinuous Galerkin finite element method for advection-reaction equations. *Comput. Methods Appl. Mech. Eng. 196*(17-20), 2071–2083.

Lacroix, V., P. Bouillard, and P. Villon (2003). An iterative defect-correction type meshless method for acoustics. *Int. J. Numer. Methods Eng. 57*(15), 2131–2146.

Ladevèze, P. and J.-P. Pelle (2005). *Mastering calculations in linear and nonlinear mechanics*. Mechanical Engineering Series. New York: Springer-Verlag. Translated from the 2001 French original by Theofanis Strouboulis.

LeVeque, R. J. (1992). *Numerical methods for conservation laws* (Second ed.). Lectures in Mathematics ETH Zürich. Basel: Birkhäuser Verlag.

Li, L., S. J. Sherwin, and P. W. Bearman (2002). A moving frame of reference algorithm for fluid/structure interaction of rotating and translating bodies. *Int. J. Numer. Methods Fluids 38*(2), 187–206.

Löhner, R., K. Morgan, and O. C. Zienkiewicz (1985). An adaptive finite element procedure for compressible high speed flows. *Comput. Methods Appl. Mech. Eng. 51*(1–3), 441–465.

MacCamy, R. C. and R. A. Fuchs (1954). Wave forces on piles: a diffraction theorys. Technical report, US Army Corps of Engineering, Beach Erosion Board, Washington, DC.

Melenk, J. and I. Babuška (1996). The partition of unity finite element method: Basic theory and applications. *Comput. Meth. Appl. Mech. Eng. 139*(1–4), 289–314.

Michoski, C., C. Mirabito, C. Dawson, D. Wirasaet, E. Kubatko, and J. Westerink (2011a). Adaptive hierarchic transformations for dynamically p-enriched slope-limiting over discontinuous Galerkin systems of generalized equations. *J. Comput. Phys. 230*(22), 8028–8056.

Michoski, C., C. Mirabito, C. Dawson, D. Wirasaet, E. Kubatko, and J. Westerink (2011b). Dynamic p-enrichment schemes for multicomponent reactive flows. *Adv. Water Resour. 34*(12), 1666–1680.

Modesto, D., G. Giorgiani, S. Zlotnik, and A. Huerta (2012). Efficiency and accuracy of high-order computations and reduced order modelling in coastal engineering wave propagation problems. In *Proceedings of the ECCOMAS Young Investigators Conferences 2012*, Aveiro, Portugal.

Montlaur, A., S. Fernández-Méndez, and A. Huerta (2008). Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. *Int. J. Numer. Methods Fluids 57*(9), 1071–1092.

Montlaur, A., S. Fernández-Méndez, J. Peraire, and A. Huerta (2010). Discontinuous Galerkin methods for the Navier-Stokes equations using solenoidal approximations. *Int. J. Numer. Methods Fluidss 64*(5), 549–564.

Nguyen, N., J. Peraire, and B. Cockburn (2010). A hybridizable discontinuous Galerkin method for Stokes flow. *Comput. Methods Appl. Mech. Eng. 199*(9-12), 582–597.

Nguyen, N. C. and J. Peraire (2011a). An adaptive shock-capturing HDG method for compressible flows. In *In Proceedings of the 20th AIAA Computational Fluid Dynamics Conference*, AIAA-2011, Honolulu, Hawaii,.

Nguyen, N. C. and J. Peraire (2011b). Navier-Stokes solution using hybridizable discontinuous Galerkin methods. In *In Proceedings of the 20th AIAA Computational Fluid Dynamics Conference*, AIAA-2011, Honolulu, Hawaii,.

Nguyen, N. C., J. Peraire, and B. Cockburn (2009). An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. *J. Comput. Phys. 228*(9), 3232–3254.

Nguyen, N. C., J. Peraire, and B. Cockburn (2011a). High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *J. Comput. Phys. 230*(10), 3695–3718.

Nguyen, N. C., J. Peraire, and B. Cockburn (2011b). Hybridizable discontinuous Galerkin methods for the time-harmonic Maxwell's equations. *J. Comput. Phys. 230*(19), 7151–7175.

Nguyen, N. C., J. Peraire, and B. Cockburn (2011c). An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations. *J. Comput. Phys. 230*(4), 1147–1170.

Oliveira, F. S. B. F. (2001). Improvement on open boundaries on a time dependent numerical model of wave propagation in the nearshore region. *Ocean Eng. 28*(1), 95–115.

Oñate, E., J. Arteaga, J. Garcia, and R. Flores (2006). Error estimation and mesh adaptivity in incompressible viscous flows using a residual power approach. *Comput. Methods Appl. Mech. Eng. 195*(4-6), 339–362.

Panchang, V. G., B. R. Pearce, G. Wei, and B. Cushman-Roisin (1991). Solution of the mild-slope wave problem by iteration. *Appl. Ocean Res. 13*(4), 187–199.

Parés, N., J. Bonet, A. Huerta, and J. Peraire (2006). The computation of bounds for linear-functional outputs of weak solutions to the two-dimensional elasticity equations. *Comput. Methods Appl. Mech. Engrg. 195*(4-6), 406–429.

Parés, N., P. Díez, and A. Huerta (2006). Subdomain-based flux-free a posteriori error estimators. *Comput. Methods Appl. Mech. Engrg. 195*(4-6), 297–323.

Parés, N., P. Díez, and A. Huerta (2008). Bounds of functional outputs for parabolic problems. II. Bounds of the exact solution. *Comput. Methods Appl. Mech. Eng. 197*(19-20), 1661–1679.

Parés, N., P. Díez, and A. Huerta (2009). Exact bounds for linear outputs of the advection-diffusion-reaction equation using flux-free error estimates. *SIAM J. Sci. Comput. 31*(4), 3064–3089.

Peraire, J., N. C. Nguyen, and B. Cockburn (2010). A hybridizable discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations. In *In Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2010, Orlando, Florida,.

Peraire, J. and P. O. Persson (2008). The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM J. Sci. Comput. 30*(4), 1806–1824.

Prudhomme, S. and J. Oden (1999). A posteriori error estimation and error control for finite element approximations of the time-dependent Navier-Stokes equations. *Finite Elem. Anal. Des. 33*(4), 247–262.

Prudhomme, S. and J. T. Oden (2003). Computable error estimators and adaptive techniques for fluid flow problems. *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics 25*, 207–268.

Reed, W. and T. Hill (1973). Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Laboratory, ASCE: Vancouver, Canada.

Remacle, J.-F., J. E. Flaherty, and M. S. Shephard (2003). An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Rev. 45*(1), 53–72.

Remacle, J.-F., S. S. Frazão, X. Li, and M. S. Shephard (2006). An adaptive discretization of shallow-water equations based on discontinuous Galerkin methods. *Int. J. Numer. Methods Fluids 52*(8), 903–923.

Roca, X., A. Gargallo-Peiró, and J. Sarrate (2011). Defining quality measures for high-order planar triangles and curved mesh generation. In *Proceedings of the 20th International Meshing Roundtable*, Paris, France.

Roca, X., N. C. Nguyen, J. Peraire, and A. Huerta (2011). A GPU-accelerated iterative solver for a hybridized discontinuous Galerkin method. In *Proceedings of the 23rd International Conference on Parallel Computational Fluid Dynamics*, Barcelona, Spain.

Sarrate, J., X. Roca, and E. Ruiz-Girones (2011). EZ4U Mesh generation environment. `www.lacan.upc.edu/ez4u.htm`.

Sauer-Budge, A. M., J. Bonet, A. Huerta, and J. Peraire (2004). Computing bounds for linear functionals of exact weak solutions to Poisson's equation. *SIAM J. Numer. Anal. 42*(4), 1610–1630.

Sevilla, R., O. Hassan, and K. Morgan (2013). An analysis of the performance of a high-order stabilised finite element method for simulating compressible flows. *Comput. Methods Appl. Mech. Eng. 253*(0), 15–27.

Smith, R. and T. Sprinks (1975). Scattering of surface waves by a conical island. *J. Fluid Mech. 72*(2), 373–384.

Steffens, L. M. and P. Díez (2009). A simple strategy to assess the error in the numerical wave number of the finite element solution of the Helmholtz equation. *Comput. Methods Appl. Mech. Eng. 198*(15-16), 1389–1400.

Steffens, L. M., N. Parés, and P. Díez (2011). Estimation of the dispersion error in the numerical wave number of standard and stabilized finite element approximations of the Helmholtz equation. *Int. J. Numer. Methods Eng. 86*(10), 1197–1224.

Steffens, L. M., N. Parés, and P. Díez (2011). Goal-oriented h-adaptivity for the Helmholtz equation: error estimates, local indicators and refinement strategies. *Comput. Mech. 47*(6), 681–699.

Stewart, J. R. and T. J. Hughes (1996). A posteriori error estimation and adaptive finite element computation of the Helmholtz equation in exterior domains. *Finite Elements in Analysis and Design 22*(1), 15–24.

Stewart, J. R. and T. J. Hughes (1997). h-adaptive finite element computation of time-harmonic exterior acoustics problems in two dimensions. *Comput. Methods Appl. Mech. Eng. 146*(1–2), 65–89.

Strouboulis, T. and R. Hidajat (2006). Partition of unity method for Helmholtz equation: q-convergence for plane-wave and wave-band local bases. *Appl. Math. 51*(2), 181–204.

Suh, K. D., C. Lee, and W. S. Park (1997). Time-dependent equations for wave propagation on rapidly varying topography. *Coast. Eng. 32*(2–3), 91–117.

Taylor, M. A., B. A. Wingate, and R. E. Vincent (2000). An algorithm for computing Fekete points in the triangle. *SIAM J. Numer. Anal 38*, 1707–1720.

Témam, R. (1969). Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (ii). *Archive for Rational Mechanics and Analysis 33*, 377–385.

Thompson, L. L. and P. M. Pinsky (1994). Complex wavenumber Fourier analysis of the p-version finite element method. *Comput. Mech. 13*(4), 255–275.

Turkel, E. and A. Yefet (1998). Absorbing PML boundary layers for wave-like equations. *Appl. Numer. Math. 27*(4), 533–557.

van Rees, W. M., A. Leonard, D. Pullin, and P. Koumoutsakos (2011). A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. *J. Comput. Phys. 230*(8), 2794–2805.

Vos, P. E. J., S. J. Sherwin, and R. M. Kirby (2010). From h to p efficiently: implementing finite and spectral/hp element methods to achieve optimal performance for low- and high-order discretisations. *J. Comput. Phys. 229*(13), 5161–5181.

Wandzurat, S. and H. Xiao (2003). Symmetric quadrature rules on a triangle. *Comput. Math. Appl. 45*(12), 1829–1840.

Wang, C. Y. (1991). Exact solutions of the steady-state Navier-Stokes equations. *Annual Review of Fluid Mechanics 23*(1), 159–177.

Wang, L. and D. J. Mavriplis (2009). Adjoint-based h-p adaptive discontinuous Galerkin methods for the 2D compressible Euler equations. *J. Comput. Phys. 228*, 7643–7661.

Wu, J., J. Z. Zhu, J. Szmelter, and O. C. Zienkiewicz (1990). Error estimation and adaptivity in Navier-Stokes incompressible flows. *Comput. Mech. 6*(4), 259–270.

Zienkiewicz, O. C., B. Boroomand, and J. Z. Zhu (6 July 1999). Recovery procedures in error estimation and adaptivity Part I: Adaptivity in linear problems. *Comput. Methods Appl. Mech. Eng. 176*(1-4), 111–125.

Zienkiewicz, O. C., R. Taylor, and P. Nithiarasu (2005). *The Finite Element Method for Fluid Dynamics* (Sixth ed.). Jordan Hill, Oxford: Elsevier Butterworth-Heinemann.

Zienkiewicz, O. C., R. L. Taylor, and J. Z. Zhu (2005). *The Finite Element Method: Its Basis & Fundamentals* (Sixth ed.). Elsevier Science.

Zienkiewicz, O. C. and J. Z. Zhu (1987). A simple error estimator and adaptive procedure for practical engineerng analysis. *Int. J. Numer. Methods Eng. 24*(2), 337–357.

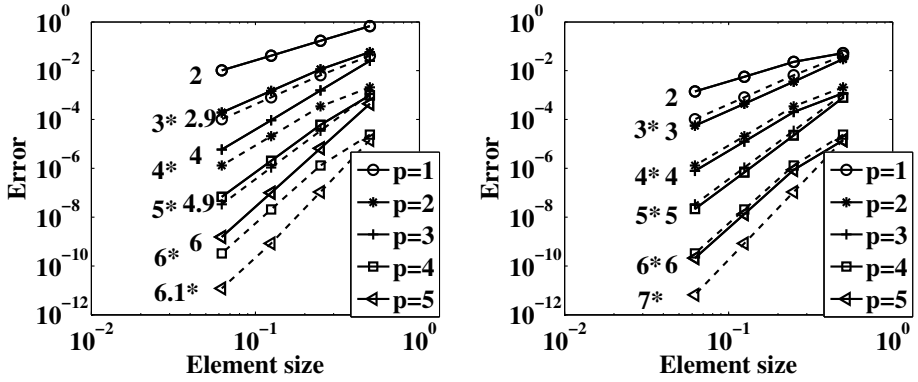
Appendix A

The influence of the τ parameter on the HDG convergence properties

In this appendix is presented a study on the influence of the stabilization parameter τ on the convergence properties of the HDG method. Two possible choices are considered: the *all faces* approach, studied also by Cockburn et al. (2009) and by Kirby et al. (2011), where τ is set to a positive constant value in all the faces of each element, and the *single face* approach, introduced by Cockburn et al. (2008), where τ is set to zero in all the faces except one, arbitrary chosen, face of each element, see (2.17). Extensive numerical evidences in the cited works show that, in both cases, values of order one provide optimal behavior for a dimensionless problem. The goal of this study is to provide indications for the choice of τ in real engineering cases, both in wave problems and fluid problems, to achieve the following two objectives:

- the coefficient τ provides enough stabilization;
- the postprocess solution is superconvergent.

In the first numerical test, the Laplace equation (2.1) is solved in the domain $\Omega =]0,1[\times]0,1[$. The function f and the boundary conditions are chosen to obtain an analytical solution $u = \cos(\pi x)\cos(\pi y)$. Nested triangular meshes of element size $h = 1/2^n$, with $n = 1 \cdots 4$ are considered, and polynomial interpolations $p = 1 \cdots 5$



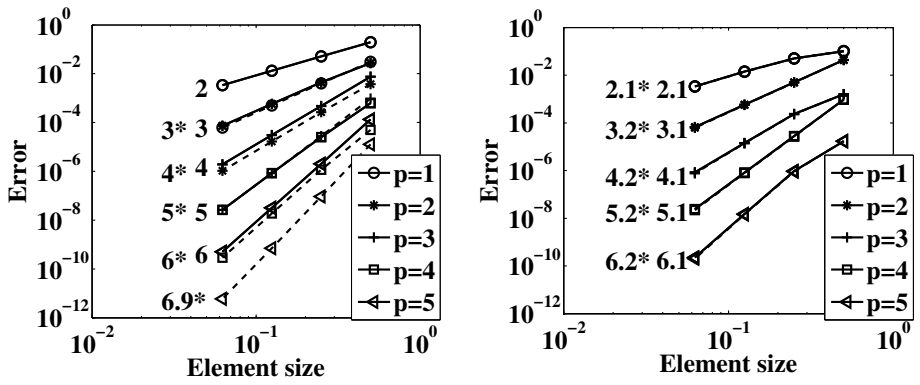
(a) Single face: $\tau_i = 1$ (b) Single face: $\tau_i = 1000$

Figure A.1: Convergence curves for the Laplace equation with *single face* stabilization parameter.

are used.

Figure A.1 and A.2 show the convergence of the $\mathcal{L}^2$ error of the HDG solution and the postprocessed solution. The results for the single face parameter are shown in Figure A.1, for of $\tau_i = 1$ and $\tau_i = 1000$, while Figure A.2 depicts the results for the all faces approach. At the bottom of each curve, the value of the final slope of the curve is also displayed. The asterisk refers to the final slope of the postprocessed solution curve. The results highlight that imposing τ different from zero in all the faces of each element can lead to loss of superconvergence, if the chosen parameter is too large. In this case, in fact, for τ_i going to infinity HDG behaves as CG with static condensation, see Kirby et al. (2011) for details. On the other hand, the superconvergent properties of the postprocess solution are conserved increasing τ_i with the single face approach.

A wave problem is considered next. The test chosen in this case is the scattering of a plane wave by a cylindrical object of unitary radius introduced in Section 2.2.3. In this case, a wavelength $\ell = 1.56$ is used. Computational meshes of element size $h = 1.6/2^n$, with $n = 1 \cdots 4$, and $p = 1 \cdots 5$ are considered. Figure A.3 and A.4 show the convergence of the error in $\mathcal{L}^2$ norm computed in the whole domain (except the PML), for the approximated solution and the postprocessed solution. The behavior is similar to the one obtained for the Laplace equation. In this case, however, the

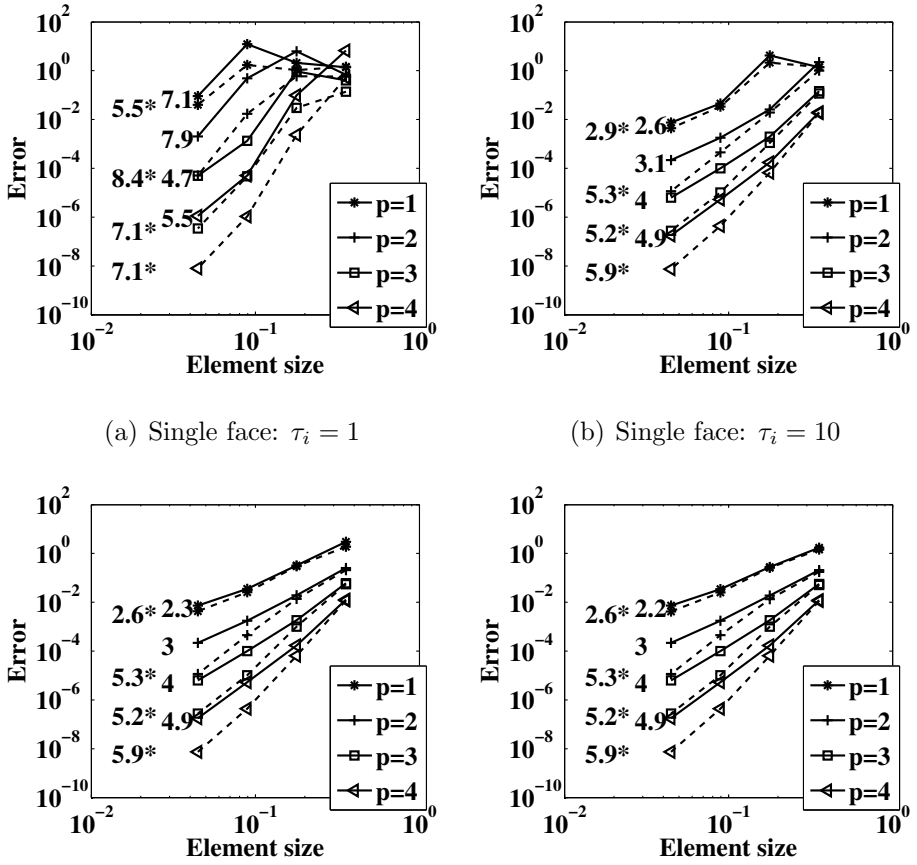


(a) All faces: $\tau_i = 1$ (b) All faces: $\tau_i = 1000$

Figure A.2: Convergence curves for the Laplace equation with *all faces* stabilization parameter.

value $\tau_i = 1$ is not sufficient, in the single face case, to provide stability for large elements. In fact, using equation (2.18), the minimum value to be used is $\tau_i \sim 50$, which would correspond to a unitary τ_i in a dimensionless problem. Hence, as expected, the convergence behavior improves as the parameter τ_i increases, reaching a constant behavior when the parameter is greater than the one prescribed by (2.18). Also for wave problems, increasing τ_i does not affect the superconvergence in case of the single face choice. In case of the all faces choice, increasing τ_i leads to loss of superconvergence, see Figure A.4. Note that, in case of non-constant coefficients in the equation, such as in (2.11), this induces an extra difficulty, because τ_i should vary in the domain to account for the variability of the coefficients.

In order to account for p-variable computations, a study of the performance of the method varying τ_i with the polynomial order is also presented. A constant discretization is then used, with $h = 0.8$, and polynomial degree $p = 1 \cdots 10$. In Figure A.5, the convergence of the postprocessed solution is shown, for three different laws of variation of τ_i with the polynomial degree. Results show that no improvements are achieved varying τ_i with p. Thus, this result show that τ_i can be independent form the actual p_{Γ_i} used on each face of the mesh.



(a) Single face: $\tau_i = 1$

(b) Single face: $\tau_i = 10$

(c) Single face: $\tau_i = 100$

(d) Single face: $\tau_i = 1000$

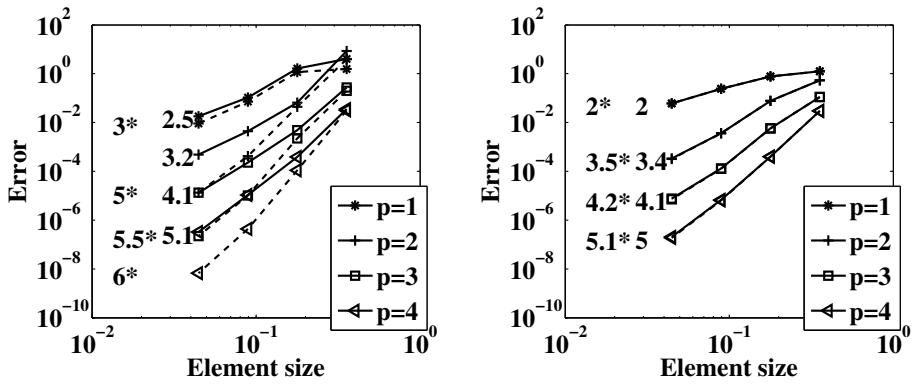
Figure A.3: Convergence curves for a scattering problem with *single face* stabilization parameter.

For fluid problems, indications on the choice of the stabilization parameter can be found in Nguyen et al. (2009, 2011c). Due to the convective character of the equations, the single face stabilization can not be used. To stabilize the convective effect of the Navier-Stokes equations, a stabilization tensor is introduced by Nguyen et al. (2011c),

$$\mathbf{S} = \tau \mathbf{I} \tag{A.1}$$

where τ is chosen as

$$\tau \approx \nu/l + |\boldsymbol{u}| \tag{A.2}$$



(a) All faces: $\tau_i = 1$ (b) All faces: $\tau_i = 1000$

Figure A.4: Convergence curves for a scattering problem with *all faces* stabilization parameter.

being ν the cinematic viscosity, l a characteristic length and $|\boldsymbol{u}|$ the velocity module. However, a direct dependence of $\mathbf{S}$ on the solution can lead to a cumbersome implementation. For the sake of simplicity, a constant scalar stabilization parameter is chosen in this work, that is $\mathbf{S} = \tau\mathbf{I}$, where τ is a positive constant set to $\tau \approx \max_\Omega(|\boldsymbol{u}|)$, i.e. of the order magnitude of the expected maximum value of the velocity in the domain. This choice provides a simple and effective criterion to tune the stabilization tensor. It must be taken into account that, with this choice, the superconvergent postprocess can be deteriorated in the area of the domain where the velocity is very low. However, in this work the main goal of the superconvergent postprocess is to provide error estimation. The definition of the error, (3.2), is based on the difference between the absolute value of the velocity and the postprocessed velocity. Hence, elements with very low velocity are less involved by the refinement process. For this reason, the deterioration of the error estimation is elements with low velocity has low impact on the final solution.

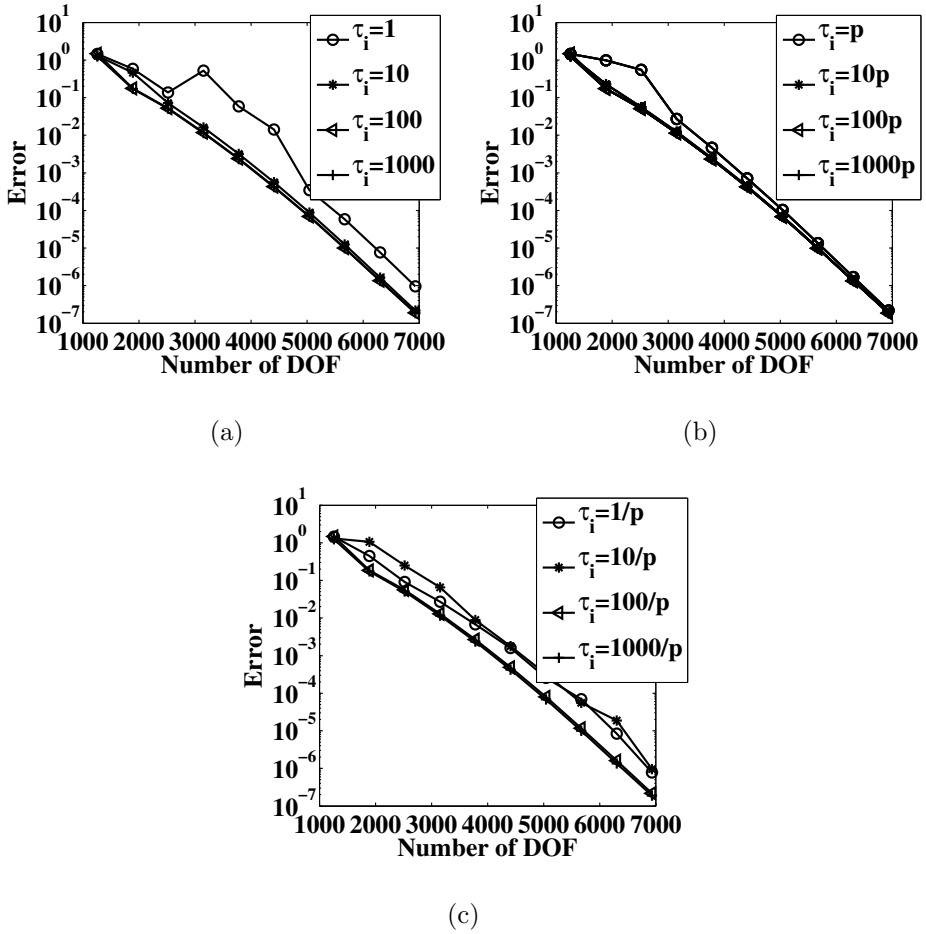


Figure A.5: Convergence curves for a scattering problem with *single face* stabilization parameter.

Appendix B

Implementation of HDG for the Mild Slope equation in frequency domain

In this Appendix are given some indications on the implementation of the HDG method for the Mild Slope equation in frequency domain, studied in Sections 2.2 and 3.1. Recall system (2.16),

$$\left.\begin{aligned} \left(\mathbf{Q}\boldsymbol{q}_h, \boldsymbol{v}\right)_{\Omega_K} + \left(u_h, \boldsymbol{\nabla}\cdot\boldsymbol{v}\right)_{\Omega_K} - \left\langle \hat{u}_h, \boldsymbol{v}\cdot\boldsymbol{n}\right\rangle_{\partial\Omega_K} &= 0 \\ \left(\boldsymbol{\nabla}\cdot\boldsymbol{q}_h, v\right)_{\Omega_K} + \left(\gamma u_h, v\right)_{\Omega_K} - \left\langle \tau(u_h-\hat{u}_h), v\right\rangle_{\partial\Omega_K} &= \left(s_x s_y f, v\right)_{\Omega_K} \end{aligned}\right\} \text{ for } K = 1,\ldots,\mathtt{n_{el}}, \tag{2.16a}$$

$$\sum_{K=1}^{\mathtt{n_{el}}} \left\langle \left(\boldsymbol{q}_h\cdot\boldsymbol{n} - \tau(u_h-\hat{u}_h)\right), \mu\right\rangle_{\partial\Omega_K} - I\left\langle k\alpha\beta\,\hat{u}_h, \mu\right\rangle_{\Gamma_{\mathtt{R}}} - I\left\langle k\beta\,\hat{u}_h, \mu\right\rangle_{\Gamma_{\mathtt{PML}}} = \left\langle g, \mu\right\rangle_{\Gamma_{\mathtt{R}}}, \tag{2.16b}$$

representing the weak form of the HDG method for the Mild Slope equation. On one hand, (2.16a) is a local system in each element Ω_K that does not involve unknowns of neighboring elements. Thus, (2.16a) can be solved element-by-element to express $\boldsymbol{q}_h$ and u_h as functions of $\hat{u}_h$. On the other hand, Equation (2.16b) is a global equation

coupling variables of different elements. Replacing $\boldsymbol{q}_h$ and u_h, solution of (2.16a) in terms of $\hat{u}_h$, in (2.16b) yields a global system on the whole mesh skeleton for the only variable $\hat{u}_h$. Recall that $\hat{u}_h$ is single valued in each face of the mesh. Once the global system is solved, $\boldsymbol{q}_h$ and u_h can be recovered for each element Ω_K using (2.16a).

The unknowns u_h and $\boldsymbol{q}_h$ are represented, element by element, with 2D nodal basis function N_j, that is

$$u_h = \sum_{i=1}^{\mathrm{n_{be}}} N_i \, \{\mathbb{U}_K\}_i, \quad \boldsymbol{q}_h = \sum_{i=1}^{\mathrm{n_{be}}} N_i \mathbf{I} \, \{\mathbb{Q}_K\}_i, \tag{B.2}$$

where $\mathrm{n_{be}}$ is the number of shape functions in the element Ω_K and $\mathbf{I}$ is the identity matrix of dimension 2×2. The vectors of nodal values in the element Ω_K for the variables u_h and $\boldsymbol{q}_h$ are denoted respectively with $\mathbb{U}_K$ and $\mathbb{Q}_K$, while $\{\mathbb{U}_K\}_i$ and $\{\mathbb{Q}_K\}_i$ are the i_{th} component of respectively $\mathbb{U}_K$ and $\mathbb{Q}_K$. The unknown $\hat{u}_h$ in the face e of the element Ω_K is denoted as

$$\hat{u}_h = \sum_{i=1}^{\mathrm{n_{bf}}} \tilde{N}_i \, \{\hat{\mathbb{U}}_{K,e}\}_i, \tag{B.3}$$

where $\tilde{N}_i$ is the i_{th} 1D basis function and $\mathrm{n_{bf}}$ is the number of shape functions in the face e of element Ω_K. Hence, considering for example a triangular element, the vector of nodal values for the unknown $\hat{u}_h$ in the element Ω_K is denoted as

$$\hat{\mathbb{U}}_K = \begin{Bmatrix} \hat{\mathbb{U}}_{K,1} \\ \hat{\mathbb{U}}_{K,2} \\ \hat{\mathbb{U}}_{K,3} \end{Bmatrix}.$$

However, note that $\hat{u}_h$ is single valued in each face of the mesh. Thus, considering a face with global numbering F, say Γ_F, shared by the two elements Ω_L and Ω_R, the nodal unknowns for $\hat{u}_h$ on Γ_F is denoted as $\hat{\mathbb{U}}_F$. The situation is represented in Figure B.1: in the this example, Γ_F corresponds to the second face of Ω_L and to the first face of Ω_R. Thus, results

$$\Gamma_F = \Gamma_{L,2} = \Gamma_{R,1},$$

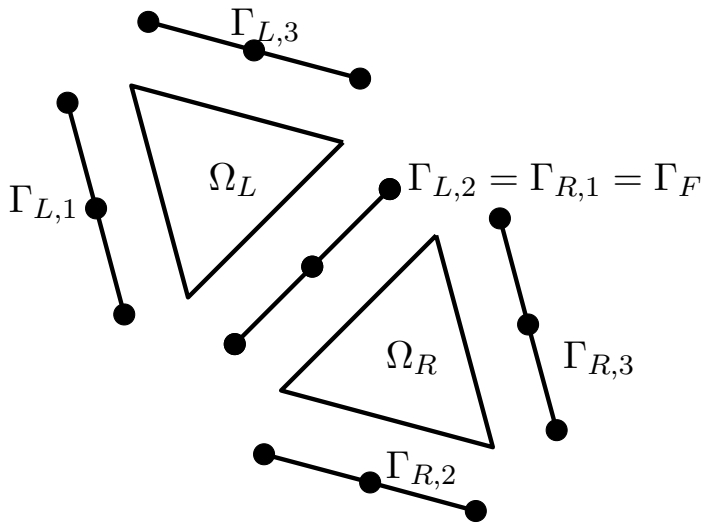


Figure B.1: Notation for the faces of two elements $p = 2$.

and therefore

$$\hat{\mathbb{U}}_F = \hat{\mathbb{U}}_{L,2} = \hat{\mathbb{U}}_{R,1}.$$

Substituting (B.2) and (B.3) in (2.16a) gives, for the generic element Ω_K, the following system of algebraic equations

$$\begin{aligned} \mathbf{A}_K \mathbb{Q}_K - \mathbf{B}_K{}^T \mathbb{U}_K &= -\mathbf{C}_{K,1}{}^T \hat{\mathbb{U}}_{K,1} - \mathbf{C}_{K,2}{}^T \hat{\mathbb{U}}_{K,2} - \mathbf{C}_{K,3}{}^T \hat{\mathbb{U}}_{K,3}, \\ \mathbf{B}_K \mathbb{Q}_K + (\mathbf{D}_K - \mathbf{M}_K) \mathbb{U}_K &= \mathbf{E}_{K,1}{}^T \hat{\mathbb{U}}_{K,1} + \mathbf{E}_{K,2}{}^T \hat{\mathbb{U}}_{K,2} + \mathbf{E}_{K,3}{}^T \hat{\mathbb{U}}_{K,3} + \mathbf{f}_K, \end{aligned} \tag{B.4}$$

where the form of the block matrices is given in Box B.1.

Solving (B.4) with respect to $\mathbb{U}_K$ and $\mathbb{Q}_K$ gives, in each element,

$$\begin{aligned} \mathbb{U}_K &= \mathbf{U}_K \, \hat{\mathbb{U}}_K + \boldsymbol{u}_K, \\ \mathbb{Q}_K &= \mathbf{Q}_K \, \hat{\mathbb{U}}_K + \boldsymbol{q}_K, \end{aligned} \tag{B.5}$$

$$
\begin{aligned}
\mathbf{A}_K^{2\mathrm{n_{be}}\times 2\mathrm{n_{be}}} &= \begin{pmatrix} \int_{\Omega_K} Q_{11} N_i N_j & 0 \\ 0 & \int_{\Omega_K} Q_{22} N_i N_j \end{pmatrix}, && \text{for } i,j = 1\cdots \mathrm{n_{be}},\\
\mathbf{B}_K^{\mathrm{n_{be}}\times 2\mathrm{n_{be}}} &= \begin{pmatrix} \int_{\Omega_K} N_i N_{j,x} & \int_{\Omega_K} N_i N_{j,y} \end{pmatrix}, && \text{for } i,j = 1\cdots \mathrm{n_{be}},\\
\mathbf{C}_{K,e}{}^{\mathrm{n_{bf}}\times 2\mathrm{n_{be}}} &= \begin{pmatrix} \int_{\Gamma_e} \tilde{N}_i N_j n_x & \int_{\Gamma_e} \tilde{N}_i N_j n_y \end{pmatrix}, && \text{for } i = 1\cdots \mathrm{n_{bf}},\ j = 1\cdots \mathrm{n_{be}},\\
\mathbf{D}_K^{\mathrm{n_{be}}\times \mathrm{n_{be}}} &= \int_{\partial\Omega_K} \tau N_i N_j, && \text{for } i,j = 1\cdots \mathrm{n_{be}},\\
\mathbf{E}_{K,e}{}^{\mathrm{n_{bf}}\times \mathrm{n_{be}}} &= \int_{\Gamma_e} \tau \tilde{N}_i N_j, && \text{for } i = 1\cdots \mathrm{n_{bf}},\ j = 1\cdots \mathrm{n_{be}},\\
\mathbf{M}_K^{\mathrm{n_{be}}\times \mathrm{n_{be}}} &= \int_{\Omega_K} N_i N_j, && \text{for } i,j = 1\cdots \mathrm{n_{be}},\\
\mathbf{f}_K^{\mathrm{n_{be}}\times 1} &= \int_{\Omega_K} s_x s_y f N_i, && \text{for } i = 1\cdots \mathrm{n_{be}}.
\end{aligned}
$$

Box B.1: Block matrices arising in the implementation of HDG for the Mild Slope equation in frequency formulation: the index $e = 1, 2, 3$ refers to the three faces of the element Ω_K, n_x and n_y are the component of the normal vector, and $N_{j,x}$ and $N_{j,y}$ are the derivatives of the shape functions in the two Cartesian directions.

having set, omitting the index K in all the terms,

$$
\begin{aligned}
\mathbf{U} &= (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{D} - \mathbf{M})^{-1}(\mathbf{E}^T + \mathbf{B}\mathbf{A}^{-1}\mathbf{C}^T),\\
\mathbf{Q} &= \mathbf{A}^{-1}[(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{D} - \mathbf{M})^{-1}(\mathbf{E}^T + \mathbf{B}\mathbf{A}^{-1}\mathbf{C}^T) - \mathbf{C}^T],\\
\boldsymbol{u} &= (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{D} - \mathbf{M})^{-1}\mathbf{f},\\
\boldsymbol{q} &= \mathbf{A}^{-1}\mathbf{B}(\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{D} - \mathbf{M})^{-1}\mathbf{f},
\end{aligned}
$$

and

$$
\mathbf{C} = \begin{Bmatrix} \mathbf{C}_{K,1} \\ \mathbf{C}_{K,2} \\ \mathbf{C}_{K,3} \end{Bmatrix}, \quad \mathbf{E} = \begin{Bmatrix} \mathbf{E}_{K,1} \\ \mathbf{E}_{K,2} \\ \mathbf{E}_{K,3} \end{Bmatrix}.
$$

Using (B.5), Equation (2.16b) allows to assembly a global system that involves only

the face unknowns for the whole faces set Γ^+. Once the the trace variable $\hat{\mathbb{U}}_F$ for $F = 1 \cdots \mathtt{n_{fa}}$ is computed, the nodal values $\mathbb{U}_K$ and $\mathbb{Q}_K$ for $K = 1 \cdots \mathtt{n_{el}}$ in the elements can be computed using System (B.5).

Substituting (B.2) in (2.16b) yields, for the interior face Γ_F shared by the elements Ω_L and Ω_R

$$\mathbf{C}_{L,e_F}\mathbb{Q}_L + \mathbf{E}_{L,e_F}\mathbb{U}_L - \mathbf{H}_F\hat{\mathbb{U}}_F + \mathbf{C}_{R,e_F}\mathbb{Q}_R + \mathbf{E}_{R,e_F}\mathbb{U}_R - \mathbf{H}_F\hat{\mathbb{U}}_F = 0, \tag{B.6}$$

where the matrices $\mathbf{C}_{L,e_F}$, $\mathbf{E}_{L,e_F}$ and $\mathbf{C}_{R,e_F}$. $\mathbf{E}_{R,e_F}$ are the ones in Box (B.1) computed on the face Γ_e corresponding to Γ_F respectively in the elements L and R, and $\mathbf{H}_F$ is

$$\mathbf{H}_F{}^{\mathrm{n_{bf}}\times\mathrm{n_{bf}}} = \{H_F\}_{ij} = \int_{\Gamma_F} \tau \tilde{N}_i \tilde{N}_j, \text{ for } i,j = 1 \cdots \mathrm{n_{bf}}.$$

Substituting (B.5) in (B.6), gives

$$\begin{aligned}(\mathbf{C}_{L,e_F}\mathbf{Q}_L + \mathbf{E}_{L,e_F}\mathbf{U}_L)\hat{\mathbb{U}}_L + (\mathbf{C}_{R,e_F}\mathbf{Q}_R + \mathbf{E}_{R,e_F}\mathbf{U}_R)\hat{\mathbb{U}}_R - 2\mathbf{H}_F\hat{\mathbb{U}}_F = \\ = -(\mathbf{C}_{L,e_F}\boldsymbol{q}_L + \mathbf{E}_{L,e_F}\boldsymbol{u}_L + \mathbf{C}_{R,e_F}\boldsymbol{q}_R + \mathbf{E}_{R,e_F}\boldsymbol{u}_R).\end{aligned} \tag{B.7}$$

Equation B.7 gives the F-th block row of the matrix of the global linear system in the unknowns $\hat{\mathbb{U}}_F$, for $F = 1 \cdots \mathtt{n_{fa}}$. Each row of the final matrix has five non-zero block entries in the positions corresponding to the five faces connected by the two elements Ω_L and Ω_R, see also Figure B.1.

For a boundary face, the boundary terms appearing in (2.16b) have to be integrated also. In this case, the block row in the final matrix has only three non-zero block entries, corresponding to the three faces of the boundary element to which the boundary face belongs.

Once solved the system obtained assembling B.7 for all the faces in the mesh, the nodal values $\mathbb{U}_K$ and $\mathbb{Q}_K$ are recovered using (B.5) with an element-by-element computation.

Appendix C

Implementation of HDG for the incompressible Navier-Stokes equations

In this Appendix are presented the main lines of the implementation of the HDG method for the Navier-Stokes equations. The block form of the elemental matrices appearing in the algebraic system is similar to those introduced for the implementation of the Mild Slope equation in Appendix B, and are not given here. The main differences between the implementation of HDG for the Navier-Stokes equations respect to the implementation for the Mild Slope equation are:

- in each element, the local problem for the pressure is closed leaving the mean of the pressure on the border of each element as a new unknown. Hence, each local problem generates a saddle point linear system, with the pressure acting as a Lagrange multiplier;

- the Navier-Stokes equations are non-linear, hence the local problems and the global problem must be solved iteratively until convergence.

The HDG weak form of the Navier-Stokes equation, with Backward-Euler time integration:

$$\left.\begin{aligned}
\big(\boldsymbol{L}_h^n, \boldsymbol{G}\big)_{\Omega_K} + \big(\boldsymbol{u}_h^n, \boldsymbol{\nabla}\cdot\boldsymbol{G}\big)_{\Omega_K} - \big\langle \hat{\boldsymbol{u}}_h^n, \boldsymbol{G}\boldsymbol{n}\big\rangle_{\partial\Omega_K} &= 0,\\
\big(\frac{\boldsymbol{u}_h^n}{\Delta t}, \boldsymbol{v}\big)_{\Omega_K} + \big(\nu\boldsymbol{L}_h^n - p_h^n\mathbf{I} - \boldsymbol{u}_h^n\otimes\boldsymbol{u}_h^n, \boldsymbol{\nabla}\boldsymbol{v}\big)_{\Omega_K} &\\
+\big\langle(-\nu\boldsymbol{L}_h^n + p_h^n\mathbf{I} + \hat{\boldsymbol{u}}_h^n\otimes\hat{\boldsymbol{u}}_h^n)\boldsymbol{n} + \mathbf{S}(\boldsymbol{u}_h^n - \hat{\boldsymbol{u}}_h^n), \boldsymbol{v}\big\rangle_{\partial\Omega_K} &=\\
\big(\boldsymbol{b}, \boldsymbol{v}\big)_{\Omega_K} + \big(\frac{\boldsymbol{u}_h^{n-1}}{\Delta t}, \boldsymbol{v}\big)_{\Omega_K},&\\
\big(\boldsymbol{u}_h^n, \boldsymbol{\nabla}v\big)_{\Omega_K} - \big\langle \boldsymbol{n}\cdot\hat{\boldsymbol{u}}_h^n, v\big\rangle_{\partial\Omega_K} &= 0,\\
\big\langle p_h^n, 1\big\rangle_{\partial\Omega_K} &= \rho_{h,K}^n,
\end{aligned}\right\} \tag{C.1a}$$

for $K = 1, \ldots, \mathtt{n_{el}}$, and

$$\begin{aligned}
\sum_i^{\mathtt{n_{el}}}\big\langle(-\nu\boldsymbol{L}_h^n + p_h^n\mathbf{I} + \hat{\boldsymbol{u}}_h^n\otimes\hat{\boldsymbol{u}}_h^n)\boldsymbol{n} + \mathbf{S}(\boldsymbol{u}_h^n - \hat{\boldsymbol{u}}_h^n), \hat{\boldsymbol{v}}\big\rangle_{\partial\Omega_K\setminus\partial\Omega_N}&\\
+\sum_i^{\mathtt{n_{el}}}\big\langle(-\nu\boldsymbol{L}_h^n + p_h^n\mathbf{I})\boldsymbol{n} + \mathbf{S}(\boldsymbol{u}_h^n - \hat{\boldsymbol{u}}_h^n), \hat{\boldsymbol{v}}\big\rangle_{\partial\Omega_N} &= 0,\\
\big\langle \boldsymbol{n}\cdot\hat{\boldsymbol{u}}_h^n, 1\big\rangle_{\partial\Omega_K} = 0,\ \text{for } K = 1, \ldots, \mathtt{n_{el}}.&
\end{aligned} \tag{C.1b}$$

System (C.1a) represents a local problem in each element Ω_K: its solution allows to express the elemental unknowns $\boldsymbol{L}_h^n, \boldsymbol{u}_h^n$ and p_h^n as a function of the trace unknown $\hat{\boldsymbol{u}}_h^n$ and the mean of the pressure on the element boundary $\rho_{h,K}^n$. Then, system (C.1b) allows to set up a global system for the unknowns $\hat{\boldsymbol{u}}_h^n$ and $\boldsymbol{\rho}_h^n$ in the whole domain.

Similarly to the procedure of Appendix B, the unknowns are represented with nodal basis functions N_i in each element Ω_K, that is

$$\boldsymbol{L}_h^n = \sum_{i=1}^{\mathtt{n_{be}}} N_i\mathbf{I}\ \{\mathbb{L}_K^n\}_i, \quad \boldsymbol{u}_h^n = \sum_{i=1}^{\mathtt{n_{be}}} N_i\mathbf{I}\ \{\mathbb{U}_K^n\}_i, \quad p_h^n = \sum_{i=1}^{\mathtt{n_{be}}} N_i\ \{\mathbb{P}_K^n\}_i, \tag{C.2}$$

where now the identity matrix $\mathbf{I}$ has dimensions $d\times d$ for the vector unknown $\boldsymbol{u}_h^n$ and $d^2\times d^2$ for the tensor unknown $\boldsymbol{L}_h^n$. The trace variable $\hat{\boldsymbol{u}}_h^n$ in the face e of the element

Ω_K is denoted as

$$\hat{\boldsymbol{u}}_h^n = \sum_{i=1}^{\mathrm{n_{bf}}} \tilde{N}_i \mathbf{I} \, \{\hat{\mathbb{U}}_{K,e}^n\}_i, \tag{C.3}$$

where $\tilde{N}_i$ is the i_{th} $d-1$-dimensional basis function and $\mathrm{n_{bf}}$ is the number of shape functions in the face. Hence, the vector of nodal values for the unknown $\hat{\boldsymbol{u}}_h^n$ in the element Ω_K is denoted as

$$\hat{\mathbb{U}}_K^n = \begin{Bmatrix} \hat{\mathbb{U}}_{K,1}^n \\ \vdots \\ \hat{\mathbb{U}}_{K,\mathtt{n_{fe}}}^n \end{Bmatrix}. \tag{C.4}$$

where $\mathtt{n_{fe}}$ is the number of faces in the element Ω_K.

Replacing the forms (C.2), (C.3) and (C.4) in system (C.1a) yields the linear system of algebraic equations in each Newton-Rapshon iteration r,

$$\begin{aligned}
\mathbf{L}_K \mathbb{L}_K^{n,r} + \mathbf{B}_K \mathbb{U}_K^{n,r} &= \mathbf{C}_K \hat{\mathbb{U}}_K^{n,r}, \\
(\mathbf{B}_K^T - \mathbf{L}_K)\mathbb{L}_K^{n,r} + (\frac{\mathbf{M}_K}{\Delta t} - \mathbf{C_v}_K^{n,r} + \mathbf{D}_K)\mathbb{U}_K^{n,r} & \\
+(\mathbf{F}_K - \mathbf{G}_K)\mathbb{P}_K^{n,r} &= (\mathbf{E}_K - \mathbf{H_v}_K^{n,r})\hat{\mathbb{U}}_K^{n,r} \\
&\quad + \frac{\mathbf{M}_K}{\Delta t}\mathbb{U}_K^{n-1} + \mathbf{f}_K^{n,r}, \\
\mathbf{R}_K \mathbb{U}_K^{n,r} - \mathbf{O}_K \hat{\mathbb{U}}_K^{n,r} &= 0, \\
\mathbf{W}_K \mathbb{P}_K^{n,r} &= \rho_{h,K}^{n,r}.
\end{aligned} \tag{C.5}$$

The convective matrices $\mathbf{C_v}_K$ and $\mathbf{H_v}_K$ derive from the discretization of the tri-linear forms related to the convective terms, that is

$$\begin{aligned}
\mathbf{C_v}_K^{n,r} &\Leftrightarrow \left(\boldsymbol{u}_h^{n,r} \otimes \boldsymbol{u}_h^{n,r-1}, \boldsymbol{\nabla v}\right)_{\Omega_K} + \left(\boldsymbol{u}_h^{n,r-1} \otimes \boldsymbol{u}_h^{n,r}, \boldsymbol{\nabla v}\right)_{\Omega_K}, \\
\mathbf{H_v}_K^{n,r} &\Leftrightarrow \left\langle (\hat{\boldsymbol{u}}^{n,r} \otimes \hat{\boldsymbol{u}}^{n,r-1})\boldsymbol{n}, \boldsymbol{v}\right\rangle_{\partial\Omega_K} + \left\langle (\hat{\boldsymbol{u}}^{n,r-1} \otimes \hat{\boldsymbol{u}}^{n,r})\boldsymbol{n}, \boldsymbol{v}\right\rangle_{\partial\Omega_K},
\end{aligned}$$

while the vector $\mathbf{f}_K$ contains the contribution of the previous time step, the body forces and the remaining part of the convective term,

$$\mathbf{f}_K^{n,r} \Leftrightarrow \big(\boldsymbol{b},\boldsymbol{v}\big)_{\Omega_K} + \big(\frac{\boldsymbol{u}_h^{n-1}}{\Delta t},\boldsymbol{v}\big)_{\Omega_K} - \big(\boldsymbol{u}_h^{n,r-1}\otimes\boldsymbol{u}_h^{n,r-1},\boldsymbol{\nabla}\boldsymbol{v}\big)_{\Omega_K} + \big\langle(\hat{\boldsymbol{u}}^{n,r-1}\otimes\hat{\boldsymbol{u}}^{n,r-1})\boldsymbol{n},\boldsymbol{v}\big\rangle_{\partial\Omega_K}.$$

System (C.5) is a constrained linear system in each element in the unknowns $\mathbb{L}_K^{n,r}, \mathbb{U}_K^{n,r}, \mathbb{P}_K^{n,r}$, at time step n and Newton-Raphson iteration r. The solution of (C.5) has the form

$$\begin{aligned}
\mathbb{L}_K^{n,r} &= \mathbf{L}_{uK}^{\;n,r}\hat{\mathbb{U}}_K^{n,r} + \mathbf{L}_{\rho K}^{\;n,r}\rho_{h,K}^{n,r} + \mathbf{L}_{0K}^{\;n,r},\\
\mathbb{U}_K^{n,r} &= \mathbf{U}_{uK}^{\;n,r}\hat{\mathbb{U}}_K^{n,r} + \mathbf{U}_{\rho K}^{\;n,r}\rho_{h,K}^{n,r} + \mathbf{U}_{0K}^{\;n,r},\\
\mathbb{P}_K^{n,r} &= \mathbf{P}_{uK}^{\;n,r}\hat{\mathbb{U}}_K^{n,r} + \mathbf{P}_{\rho K}^{\;n,r}\rho_{h,K}^{n,r} + \mathbf{P}_{0K}^{\;n,r},
\end{aligned} \tag{C.6}$$

that is, a local mapping from the nodal values of the trace unknown $\hat{\boldsymbol{u}}_h^n$ and the mean of the pressure on the element borders $\boldsymbol{\rho}_h^n$ to the nodal values of the elemental unknowns $\boldsymbol{L}_h^n$, $\boldsymbol{u}_h^n$ and p_h^n.

Once solved for the local mapping element-by-element, system (C.1b) allows to set up the global system for the unknowns $\hat{\mathbb{U}}_F^{n,r}$ for $F = 1\cdots\mathtt{n_{fa}}$ and $\rho_{h,K}^{n,r}$ for $K = 1\cdots\mathtt{n_{el}}$. Thus, substituting (C.2), (C.3) and (C.4) in (C.1b) yields, for the interior face Γ_F shared by the elements Ω_L and Ω_R

$$\begin{aligned}
-\mathbf{L}_{L,e_F}\mathbb{L}_L^{n,r} + \mathbf{D}_{L,e_F}\mathbb{U}_L^{n,r} + (\mathbf{H}_F^{n,r} - \mathbf{E}_F)\hat{\mathbb{U}}_F^{n,r} + \mathbf{F}_{L,e_F}\mathbb{P}_L^{n,r}\\
-\mathbf{L}_{R,e_F}\mathbb{L}_R^{n,r} + \mathbf{D}_{R,e_F}\mathbb{U}_R^{n,r} + (\mathbf{H}_F^{n,r} - \mathbf{E}_F)\hat{\mathbb{U}}_F^{n,r} + \mathbf{F}_{R,e_F}\mathbb{P}_R^{n,r} = 0,\\
\mathbf{Y}_L\cdot\hat{\mathbb{U}}_F^{n,r} + \mathbf{Y}_R\cdot\hat{\mathbb{U}}_F^{n,r} = 0.
\end{aligned} \tag{C.7}$$

If no Neumann boundaries are present in the domain, the problem for the pressure can be closed imposing the average of the pressure in the domain equal to zero, that is

$$\sum_{K=1}^{\mathtt{n_{el}}} \rho_{h,K}^{n,r} A_K = 0, \tag{C.8}$$

where A_K is the measure of the element Ω_K.

Substituting (C.6) in (C.7) results

$$
\begin{aligned}
&\left(-\mathbf{L}_{L,e_F}\mathbf{L}_{u_L}^{n,r}+\mathbf{D}_{L,e_F}\mathbf{U}_{u_L}^{n,r}+\mathbf{F}_{L,e_F}\mathbf{P}_{u_L}^{n,r}\right)\hat{\mathbb{U}}_L^{n,r}\\
+&\left(-\mathbf{L}_{R,e_F}\mathbf{L}_{u_R}^{n,r}+\mathbf{D}_{R,e_F}\mathbf{U}_{u_R}^{n,r}+\mathbf{F}_{R,e_F}\mathbf{P}_{u_R}^{n,r}\right)\hat{\mathbb{U}}_R^{n,r}\\
+&\left(-\mathbf{L}_{L,e_F}\mathbf{L}_{\rho_L}^{n,r}+\mathbf{D}_{L,e_F}\mathbf{U}_{\rho_L}^{n,r}+\mathbf{F}_{L,e_F}\mathbf{P}_{\rho_L}^{n,r}\right)\rho_{h,L}^{n,r}\\
+&\left(-\mathbf{L}_{R,e_F}\mathbf{L}_{\rho_R}^{n,r}+\mathbf{D}_{R,e_F}\mathbf{U}_{\rho_R}^{n,r}+\mathbf{F}_{R,e_F}\mathbf{P}_{\rho_R}^{n,r}\right)\rho_{h,R}^{n,r} \qquad \text{(C.9)}\\
+&2\left(\mathbf{H}_F^{n,r}-\mathbf{E}_F\right)\hat{\mathbb{U}}_F=\\
&\left(\mathbf{L}_{L,e_F}\mathbf{L}_{0_L}^{n,r}-\mathbf{D}_{L,e_F}\mathbf{U}_{0_L}^{n,r}-\mathbf{F}_{L,e_F}\mathbf{P}_{0_L}^{n,r}\right)+\\
&\left(\mathbf{L}_{R,e_F}\mathbf{L}_{0_R}^{n,r}-\mathbf{D}_{R,e_F}\mathbf{U}_{0_R}^{n,r}-\mathbf{F}_{R,e_F}\mathbf{P}_{0_R}^{n,r}\right),\\
&\mathbf{Y}_L\cdot\hat{\mathbb{U}}_L^{n,r}+\mathbf{Y}_R\cdot\hat{\mathbb{U}}_R^{n,r}=0. \qquad \text{(C.10)}
\end{aligned}
$$

Figure C.1 depicts the block form of the final system matrix in the nodal values of $\hat{\mathbb{U}}_F^{n,r}$, for $F=1\cdots\mathtt{n_{fa}}$, and $\boldsymbol{\rho}_h^{n,r}$. Equation (C.9) represents the F-th block row of the block matrix $\mathbb{A}$: the number of non-zero blocks in the block row is equal to the number of faces connected by the two elements Ω_L and Ω_R. On the other hand, Equation (C.10) represents the L and R rows of matrix $\mathbb{B}$, see Figure C.1.

The linear system produced by the assembly of (C.9) and (C.10) for all the faces, and represented in Figure C.1, is a saddle point system that has to be solved at each Newton-Raphson iteration. The mean of the pressure on the element boundaries $\rho_{h,K}^{n,r}$ acts as a Lagrange multiplier for the nodal values of the trace of the velocity $\hat{\mathbb{U}}_F^{n,r}$ for $F=1\cdots\mathtt{n_{fa}}$. Equation (C.8), in case of no Neumann boundary conditions in $\partial\Omega$, produces another constrain on the final system, which is also displayed in Figure C.1. Once solved the linear system for the trace of the velocity $\hat{\mathbb{U}}_F^{n,r}$ and the mean of the pressure in each element border $\boldsymbol{\rho}_h^{n,r}$, the nodal values $\mathbb{L}_K^{n,r}$, $\mathbb{U}_K^{n,r}$ and $\mathbb{P}_K^{n,r}$ are recovered using (C.6) with an element-by-element computation.

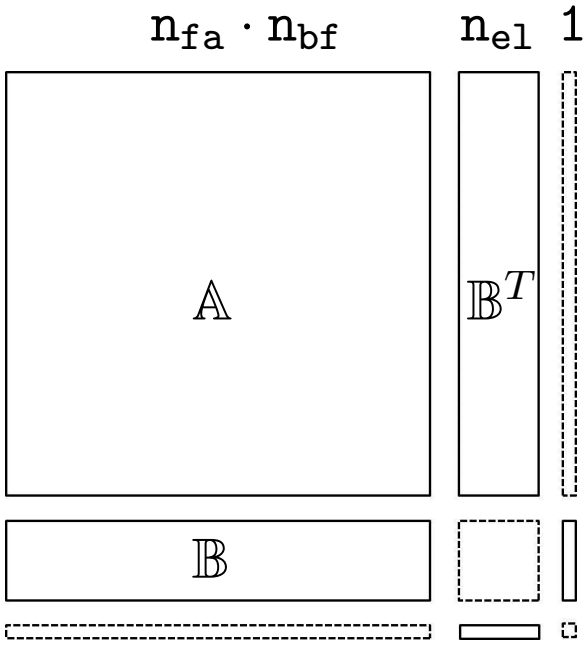


Figure C.1: Block form of the final linear system matrix for the Navier-Stokes equations. The blocks with dashed lines represent zero blocks. The last column derives from fixing the average of the pressure in the whole domain, in case of no Neumann boundary conditions are present.

Appendix D

Fractional step methods for HDG

In this section is presented the derivation of two fractional step methods for the time integration of the unsteady incompressible Navier-Stokes equations. The motivation of this work is to be able to tackle 3D problems. It is well know that in 3D the solution of the linear system becomes critical, in particular for high order elements, due to the increased number of connections between the elements respect to the 2D case. For this reason, for direct solvers, the solution time and the memory requirements to perform the LU factorization increase faster when increasing the number of spatial dimensions. This is shown in Table D.1, where the runtime and the fill-in for a Laplace problem is considered as a function of the mesh size. For iterative solvers, the runtime get smaller increasing the number of spatial dimensions. Thus, iterative solvers are more efficients than direct solvers in 3D. However, for the Navier-Stokes equations, the saddle-point nature of the linear system introduces an additional difficulty. Here is where the fractional step methods come into play.

Fractional step methods were introduced by Chorin (1968) and Témam (1969) as time integration techniques for the unsteady incompressible Navier-Stokes equations. The time advancement is decomposed into a sequence of two or more steps. In particular, in the original version, the momentum equation is solved in the first step without accounting for the diverge free constrain, obtaining in this way an intermediate velocity that is projected into a divergence free space in the second step.

Table D.1: Runtime and fill-in for direct and iterative solvers as function the dimension of the system N and the number of spatial dimensions.

	1D	**2D**	**3D**
		Direct solver	
Fill in	$\mathcal{O}(N)$	$\mathcal{O}(N \log N)$	$\mathcal{O}(N^{4/3})$
Runtime (flops)	$\mathcal{O}(N)$	$\mathcal{O}(N^{3/2})$	$\mathcal{O}(N^2)$
		Conjugate gradient no precond.	
Runtime (flops)	$\mathcal{O}(N^2)$	$\mathcal{O}(N^{3/2})$	$\mathcal{O}(N^{4/3})$

Fractional step methods are very popular in CG and finite volume schemes for fluid problems, but they have had little diffusion between DG users. Nevertheless, the uncoupling of the pressure and velocity allows to circumvent the difficulties caused by the saddle-point nature of the variational formulation of the Navier-Stokes equations, where the pressure variable acts as a Lagrange multiplier of the incompressibility constrain. In fact, the HDG spatial discretization produces an algebraic system for the nodal values of trace of the velocity and the mean of the pressure governed by a partitioned matrix with a null submatrix on the diagonal, see also Appendix C. Solving this system using iterative solvers is not trivial. Other solutions to avoid the saddle-point linear system, such as the Augmented Lagrangian approach proposed by Nguyen et al. (2011c), do not provide a robust solution for high Reynolds numbers, see Section D.3.

In the fractional step method, in each step a simpler problem is solved. The HDG discretization of each step produces a non-constrained linear system which can be easily solved with preconditioned iterative methods. In the following, the HDG fractional step methods of first and second-order are presented, and some numerical tests are proposed to show the accuracy and the computational efficiency of the schemes.

D.1 First-order scheme

Starting from the Navier-Stokes equations (2.20), the implicit Euler time discretization of first-order gives

$$\begin{aligned} \boldsymbol{L}^n - \boldsymbol{\nabla}\boldsymbol{u}^n &= \boldsymbol{0}, && \text{in } \Omega, \\ \frac{\boldsymbol{u}^n - \boldsymbol{u}^{n-1}}{\Delta t} + \boldsymbol{\nabla}\cdot(\boldsymbol{u}^n \otimes \boldsymbol{u}^n - \nu\boldsymbol{L}^n + \boldsymbol{\nabla}p^n\mathbf{I}) &= \boldsymbol{b}^n && \text{in } \Omega, \\ \boldsymbol{\nabla}\cdot\boldsymbol{u}^n &= 0 && \text{in } \Omega, \\ \boldsymbol{u}^n &= \boldsymbol{g} && \text{on } \partial\Omega_D, \\ (-p^n\mathbf{I} + \nu\boldsymbol{L}^n)\boldsymbol{n} &= \boldsymbol{0} && \text{on } \partial\Omega_N. \end{aligned} \tag{D.1}$$

The first-order projection scheme is based on the approximate equivalence of (D.1) with the two following systems

- **1st step**

$$\begin{cases} \tilde{\boldsymbol{L}}^n - \boldsymbol{\nabla}\tilde{\boldsymbol{u}}^n = \boldsymbol{0}, & \text{in } \Omega, \\ \dfrac{\tilde{\boldsymbol{u}}^n - \boldsymbol{u}^{n-1}}{\Delta t} + \boldsymbol{\nabla}\cdot(\tilde{\boldsymbol{u}}^n \otimes \tilde{\boldsymbol{u}}^n - \nu\tilde{\boldsymbol{L}}^n) = \boldsymbol{b}^n & \text{in } \Omega, \\ \tilde{\boldsymbol{u}}^n = \boldsymbol{g} & \text{on } \partial\Omega_D, \\ \nu\tilde{\boldsymbol{L}}^n\boldsymbol{n} = \boldsymbol{0} & \text{on } \partial\Omega_N, \end{cases} \tag{D.2}$$

- **2nd step**

$$\begin{cases} \dfrac{\boldsymbol{u}^n - \tilde{\boldsymbol{u}}^n}{\Delta t} + \boldsymbol{\nabla}p^n = \boldsymbol{0} & \text{in } \Omega, \\ \boldsymbol{\nabla}\cdot\boldsymbol{u}^n = 0, & \text{in } \Omega, \\ \boldsymbol{n}\cdot\boldsymbol{u}^n = \boldsymbol{n}\cdot\boldsymbol{g} & \text{on } \partial\Omega_D, \\ p^n\mathbf{I}\boldsymbol{n} = \boldsymbol{0} & \text{on } \partial\Omega_N. \end{cases} \tag{D.3}$$

where $\tilde{\boldsymbol{u}}^n$ is called the intermediate velocity, while $\boldsymbol{u}^n$ and p^n are now called the end-of-step variables. Details on the error introduced in the approximation of (D.1) with (D.2) and (D.3) can be found in Codina (2001). Particular care must be taken for

the treatment of the boundary condition in the projection scheme: on the Dirichlet boundary, in the first step the intermediate velocity $\tilde{\boldsymbol{u}}$ is set equal to the prescribed velocity $\boldsymbol{g}$, while in the second step, only the normal component can be imposed. On the Neumann boundary, the boundary condition is decomposed to include the viscous part of the prescribed boundary traction in the first step and the pressure contribution in the second-step. Details on the imposition of the boundary condition in projection schemes can be found in Donea and Huerta (2003).

To avoid a saddle point problem in the second step, (D.3) is rewritten taking the divergence of the first equation, giving

$$\begin{aligned} \boldsymbol{\nabla}\cdot\boldsymbol{\nabla}p^n &= \frac{1}{\Delta t}\boldsymbol{\nabla}\cdot\tilde{\boldsymbol{u}}^n \quad \text{in } \Omega, \\ p^n\mathbf{I}\boldsymbol{n} &= \mathbf{0} \qquad\qquad \text{on } \partial\Omega, \end{aligned} \tag{D.4}$$

which is the *pressure-Poisson* equation for the second step. Once solved D.4 for the end-of-step pressure p^n, the end-of-step velocity $\boldsymbol{u}^n$ can be computed by the explicit relation $\boldsymbol{u}^n = \tilde{\boldsymbol{u}}^n - \Delta t\boldsymbol{\nabla}p^n$. Note that the boundary condition for the end-of-step pressure does not need to be satisfied by the unsplit original problem. This means that the splitting procedure introduces a non-physical boundary layer in the pressure profile. However, some procedure have been proposed to minimize this effect, see Guermond et al. (2004).

The HDG solution of (D.2) and (D.3) is now considered. Calling $\hat{\boldsymbol{u}}_h^n$ the trace variable of the intermediate velocity $\tilde{\boldsymbol{u}}_h^n$, the HDG problem for the first step (D.2) is: find an approximation $(\tilde{\boldsymbol{L}}_h^n, \tilde{\boldsymbol{u}}_h^n, \hat{\boldsymbol{u}}_h^n) \in [\widehat{\mathcal{V}}_h]^{d\times d} \times [\widehat{\mathcal{V}}_h]^d \times [\widehat{\Lambda}_h(\boldsymbol{g})]^d$ such that

$$\left.\begin{aligned} (\tilde{\boldsymbol{L}}_h^n, \boldsymbol{G})_{\Omega_i} + (\tilde{\boldsymbol{u}}_h^n, \boldsymbol{\nabla}\cdot\boldsymbol{G})_{\Omega_i} - \left\langle \hat{\boldsymbol{u}}_h^n, \boldsymbol{G}\boldsymbol{n}\right\rangle_{\partial\Omega_i} &= 0, \\ (\frac{\tilde{\boldsymbol{u}}_h^n}{\Delta t}, \boldsymbol{v})_{\Omega_i} + (\nu\tilde{\boldsymbol{L}}_h^n - \tilde{\boldsymbol{u}}_h^n\otimes\tilde{\boldsymbol{u}}_h^n, \boldsymbol{\nabla}\boldsymbol{v})_{\Omega_i} & \\ +\left\langle(-\nu\tilde{\boldsymbol{L}}_h^n + \hat{\boldsymbol{u}}_h^n\otimes\hat{\boldsymbol{u}}_h^n)\boldsymbol{n} + \mathbf{S}(\tilde{\boldsymbol{u}}_h^n - \hat{\boldsymbol{u}}_h^n), \boldsymbol{v}\right\rangle_{\partial\Omega_i} &= (\boldsymbol{b}, \boldsymbol{v})_{\Omega_i} \\ &\quad + (\frac{\boldsymbol{u}_h^{n-1}}{\Delta t}, \boldsymbol{v})_{\Omega_i} \end{aligned}\right\} \tag{D.5a}$$

for $i = 1, \dots, \mathtt{n_{el}}$, and

$$\begin{aligned}\sum_{i}^{\mathtt{n_{el}}} \big\langle (-\nu \tilde{\boldsymbol{L}}_h^n + \hat{\boldsymbol{u}}_h^n \otimes \hat{\boldsymbol{u}}_h^n)\boldsymbol{n} + \mathbf{S}(\tilde{\boldsymbol{u}}_h^n - \hat{\boldsymbol{u}}_h^n), \boldsymbol{\mu} \big\rangle_{\partial\Omega_i \setminus \partial\Omega_N} \\ + \sum_{i}^{\mathtt{n_{el}}} \big\langle -\nu \tilde{\boldsymbol{L}}_h^n \boldsymbol{n} + \mathbf{S}(\tilde{\boldsymbol{u}}_h^n - \hat{\boldsymbol{u}}_h^n), \boldsymbol{\mu} \big\rangle_{\partial\Omega_i \bigcap \partial\Omega_N} = 0.\end{aligned} \tag{D.5b}$$

for all $(\boldsymbol{G}, \boldsymbol{v}, \boldsymbol{\mu}) \in [\widehat{\mathcal{V}}_h]^{d\times d} \times [\widehat{\mathcal{V}}_h]^d \times [\widehat{\Lambda}_h(\mathbf{0})]^d$. Similarly to the unsplit problem, system (D.5a) is a local problem in each element in the variables $\tilde{\boldsymbol{L}}_h^n$ and $\tilde{\boldsymbol{u}}_h^n$, to be solved as a function of the trace variable $\hat{\boldsymbol{u}}_h^n$. Once solved (D.5a), the global problem (D.5b) allows to set up a linear system in the only unknown $\hat{\boldsymbol{u}}_h^n$. No saddle point is present in this linear system. The intermediate velocity $\tilde{\boldsymbol{u}}_h^n$ is then computed element-by-element once obtained $\hat{\boldsymbol{u}}_h^n$ in the mesh faces.

In the second step, the intermediate velocity is projected into the divergence free space. This is performed solving first for end-of-step pressure variable p_h^n using the pressure-Poisson problem. Introducing the new variable $\boldsymbol{r} = -\boldsymbol{\nabla} p$, and following the procedure described in Section 2.1, the HDG discretization of the pressure-Poisson problem is: find $(\boldsymbol{r}_h^n, p_h^n, \hat{p}_h^n) \in [\widehat{\mathcal{V}}_h]^d \times \widehat{\mathcal{V}}_h \times \widehat{\Lambda}_h$ such that

$$\left.\begin{aligned}(\boldsymbol{r}_h^n, \boldsymbol{v})_{\Omega_i} - (p_h^n, \boldsymbol{\nabla}\cdot\boldsymbol{v})_{\Omega_i} + \big\langle \hat{p}_h^n, \boldsymbol{v}\cdot\boldsymbol{n} \big\rangle_{\partial\Omega_i} = 0, \\ (\boldsymbol{\nabla}\cdot\boldsymbol{r}_h^n, v)_{\Omega_i} + \big\langle \tau(p_h^n - \hat{p}_h^n), v \big\rangle_{\partial\Omega_i} = -\frac{1}{\Delta t}(\boldsymbol{\nabla}\cdot\tilde{\boldsymbol{u}}_h^n, v)_{\Omega_i},\end{aligned}\right\} \tag{D.6a}$$

for $i = 1, \dots, \mathtt{n_{el}}$, and

$$\sum_{i=1}^{\mathtt{n_{el}}} \langle \boldsymbol{r}_h^n \cdot \boldsymbol{n} + \tau(p_h^n - \hat{p}_h^n), \mu \rangle_{\partial\Omega_i} = 0, \tag{D.6b}$$

for all $(\boldsymbol{v}, v, \mu) \in [\widehat{\mathcal{V}}_h]^d \times \widehat{\mathcal{V}}_h \times \widehat{\Lambda}_h$, where $\hat{p}_h$ represents the trace variable of the pressure on the element boundaries. Again, the HDG solution of the pressure-Poisson problem involves solving a local problem (D.6a) in the unknowns $\boldsymbol{r}_h^n$ and p_h^n as a function of the trace of the pressure $\hat{p}_h^n$. The global problem (D.6b) is then solved for the trace of the pressure in the whole mesh skeleton. Particular attention must be paid for

the solution of the linear system produced by (D.6): in fact, since only Neumann boundary conditions are present in (D.6), the matrix governing this linear system is singular. Here, the solution is guaranteed providing that the right hand side of the linear system belongs to the columns space of the matrix. Anyway, since the pressure is a scalar variable, this linear system is smaller then the one produced by the first step, even if equal polynomial interpolation are used for the velocity and pressure variables. Thus, the projection step is a cheap computation, in particular for 3D problems.

Once obtained the end-of-step pressure p_h^n and its derivative $\boldsymbol{r}_h^n$ in each element , the end-of-step velocity is determined with the relation $\boldsymbol{u}_h^n = \tilde{\boldsymbol{u}}_h^n + \Delta t \boldsymbol{r}_h^n$.

D.2 Second-order scheme

The introduction of a pressure term in the first step (D.2) improves the accuracy of the time integration. This scheme is also called *incremental pressure-correction scheme*. Using the Backward Difference Formula of second-order (BDF2) to approximate the time derivative, the two steps of the scheme are:

- **1st step**

$$\begin{cases} \tilde{\boldsymbol{L}}^n - \boldsymbol{\nabla}\tilde{\boldsymbol{u}}^n = \boldsymbol{0}, & \text{in } \Omega, \\ \dfrac{3\tilde{\boldsymbol{u}}^n - 4\tilde{\boldsymbol{u}}^{n-1} + \tilde{\boldsymbol{u}}^{n-2}}{2\Delta t} & \\ +\boldsymbol{\nabla}\cdot(\tilde{\boldsymbol{u}}^n \otimes \tilde{\boldsymbol{u}}^n - \nu\tilde{\boldsymbol{L}}^n + p^{n-1}\mathbf{I}) = \boldsymbol{b}^n & \text{in } \Omega, \\ \tilde{\boldsymbol{u}}^n = \boldsymbol{g} & \text{on } \partial\Omega_D, \\ \nu\tilde{\boldsymbol{L}}^n\boldsymbol{n} = \boldsymbol{0} & \text{on } \partial\Omega_N, \end{cases} \tag{D.7}$$

- **2nd step**

$$\begin{cases} \dfrac{3\boldsymbol{u}^n - 3\tilde{\boldsymbol{u}}^n}{2\Delta t} + \boldsymbol{\nabla}(p^n - p^{n-1}) = \boldsymbol{0} & \text{in } \Omega, \\ \boldsymbol{\nabla}\cdot\boldsymbol{u}^n = 0, & \text{in } \Omega, \\ \boldsymbol{n}\cdot\boldsymbol{u}^n = \boldsymbol{n}\cdot\boldsymbol{g} & \text{on } \partial\Omega_D, \\ p^n\mathbf{I}\boldsymbol{n} = \boldsymbol{0} & \text{on } \partial\Omega_N. \end{cases} \tag{D.8}$$

As for the first-order scheme, system (D.8) is replaced by the pressure-Poisson problem, which in this case is

$$\begin{aligned} \boldsymbol{\nabla}\cdot\boldsymbol{\nabla}p^n &= \frac{3}{2\Delta t}\boldsymbol{\nabla}\cdot\tilde{\boldsymbol{u}}^n + \boldsymbol{\nabla}\cdot\boldsymbol{\nabla}p^{n-1} && \text{in } \Omega, \\ p^n\mathbf{I}\boldsymbol{n} &= \mathbf{0} && \text{on } \partial\Omega. \end{aligned}$$

Using the settings of the first-order scheme, the HDG discrete problem of the first step (D.7) is: find an approximation $(\tilde{\boldsymbol{L}}_h^n, \tilde{\boldsymbol{u}}_h^n, \hat{\boldsymbol{u}}_h^n) \in [\widehat{\mathcal{V}}_h]^{d\times d} \times [\widehat{\mathcal{V}}_h]^d \times [\widehat{\Lambda}_h(\boldsymbol{g})]^d$ such that

$$\left.\begin{aligned} (\tilde{\boldsymbol{L}}_h^n, \boldsymbol{G})_{\Omega_i} + (\tilde{\boldsymbol{u}}_h^n, \boldsymbol{\nabla}\cdot\boldsymbol{G})_{\Omega_i} - \big\langle \hat{\boldsymbol{u}}_h^n, \boldsymbol{G}\boldsymbol{n}\big\rangle_{\partial\Omega_i} &= 0, \\ \frac{3}{2}(\frac{\tilde{\boldsymbol{u}}_h^n}{\Delta t}, \boldsymbol{v})_{\Omega_i} + (\nu\tilde{\boldsymbol{L}}_h^n - \tilde{\boldsymbol{u}}_h^n\otimes\tilde{\boldsymbol{u}}_h^n, \boldsymbol{\nabla}\boldsymbol{v})_{\Omega_i} & \\ + \big\langle(-\nu\tilde{\boldsymbol{L}}_h^n + \hat{\boldsymbol{u}}_h^n\otimes\hat{\boldsymbol{u}}_h^n)\boldsymbol{n} + \mathbf{S}(\tilde{\boldsymbol{u}}_h^n - \hat{\boldsymbol{u}}_h^n), \boldsymbol{v}\big\rangle_{\partial\Omega_i} &= \\ (\boldsymbol{b}, \boldsymbol{v})_{\Omega_i} + 2(\frac{\boldsymbol{u}_h^{n-1}}{\Delta t}, \boldsymbol{v})_{\Omega_i} - (\frac{\boldsymbol{u}_h^{n-2}}{\Delta t}, \boldsymbol{v})_{\Omega_i} + (\boldsymbol{r}_h^{n-1}, \boldsymbol{v})_{\Omega_i}, & \end{aligned}\right\} \text{ for } i = 1, \ldots, \mathtt{n_{el}},$$

$$\begin{aligned} \sum_i^{\mathtt{n_{el}}} \big\langle(-\nu\tilde{\boldsymbol{L}}_h^n + \hat{\boldsymbol{u}}_h^n\otimes\hat{\boldsymbol{u}}_h^n)\boldsymbol{n} + \mathbf{S}(\tilde{\boldsymbol{u}}_h^n - \hat{\boldsymbol{u}}_h^n), \boldsymbol{\mu}\big\rangle_{\partial\Omega_i\setminus\partial\Omega_N} & \\ + \sum_i^{\mathtt{n_{el}}} \big\langle -\nu\tilde{\boldsymbol{L}}_h^n\boldsymbol{n} + \mathbf{S}(\tilde{\boldsymbol{u}}_h^n - \hat{\boldsymbol{u}}_h^n), \boldsymbol{\mu}\big\rangle_{\partial\Omega_i\bigcap\partial\Omega_N} &= 0, \end{aligned}$$

for all $(\boldsymbol{G}, \boldsymbol{v}, \boldsymbol{\mu}) \in [\widehat{\mathcal{V}}_h]^{d\times d} \times [\widehat{\mathcal{V}}_h]^d \times [\widehat{\Lambda}_h(\mathbf{0})]^d$.

For the second step (D.8), the discrete HDG problem is: find $(\boldsymbol{r}_h^n, p_h^n, \hat{p}_h^n) \in [\widehat{\mathcal{V}}_h]^d \times \widehat{\mathcal{V}}_h \times \widehat{\Lambda}_h$ such that

$$\left.\begin{aligned} (\boldsymbol{r}_h^n, \boldsymbol{v})_{\Omega_i} - (p_h^n, \boldsymbol{\nabla}\cdot\boldsymbol{v})_{\Omega_i} + \big\langle\hat{p}_h^n, \boldsymbol{v}\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_i} &= 0, \\ (\boldsymbol{\nabla}\cdot\boldsymbol{r}_h^n, v)_{\Omega_i} + \big\langle\tau(p_h^n - \hat{p}_h^n), v\big\rangle_{\partial\Omega_i} &= \\ -\frac{3}{2\Delta t}(\boldsymbol{\nabla}\cdot\tilde{\boldsymbol{u}}_h^n, v)_{\Omega_i}, + (\boldsymbol{\nabla}\cdot\boldsymbol{r}_h^{n-1}, v)_{\Omega_i}, & \end{aligned}\right\} \text{ for } i = 1, \ldots, \mathtt{n_{el}},$$

$$\sum_{i=1}^{\mathtt{n_{el}}} \langle \boldsymbol{r}_h^n \cdot \boldsymbol{n} + \tau(p_h^n - \hat{p}_h^n), \mu \rangle_{\partial\Omega_i} = 0,$$

for all $(\boldsymbol{v}, v, \mu) \in [\widehat{\mathcal{V}}_h]^d \times \widehat{\mathcal{V}}_h \times \widehat{\Lambda}_h$.

D.3 Numerical tests

In this section are shown some numerical tests for the validation of the fractional step methods introduced.

The first test is aimed at checking the time accuracy of the first and second-order scheme. To this aim, the 2D Green-Taylor vortex problem, see Nguyen et al. (2011c), is solved on a square domain $\Omega \in]0,1[\times]0,1[$. Zero body forces are set and the exact solution for the velocity is

$$\boldsymbol{u} = \begin{pmatrix} -\cos(\pi x)\sin(\pi y)\exp(-2\pi^2 t/\mathrm{Re}), \\ \sin(\pi x)\cos(\pi y)\exp(-2\pi^2 t/\mathrm{Re}) \end{pmatrix}.$$

The above problem is considered with Re= 20 and final time $T = 1$.

Figure D.1 shows the time convergence of the $\mathcal{L}^2$ velocity error in the domain for the first and second-order fractional step scheme, compared to the direct solve of the saddle-point problem. Convergence rates very close to the theoretical ones are obtained, while the approximation introduced in the splitting procedure introduces in this case an error that is one order of magnitude greater respect to the direct solve. Even if this error is not negligible, it should be taken in mind that the splitting technique is aimed at solving large problems where the direct Gauss elimination is precluded either for computational or memory reasons.

The solution of the original saddle point with an iterative solver is also awkward for the difficulties related to the parameters setting for the pre-conditioner. As an example, the second numerical test compares the solution of the linear system produced by the HDG discretization of the NACA problem of Section 2.3.2 with $p = 4$ and Re= 1000 with different approaches:

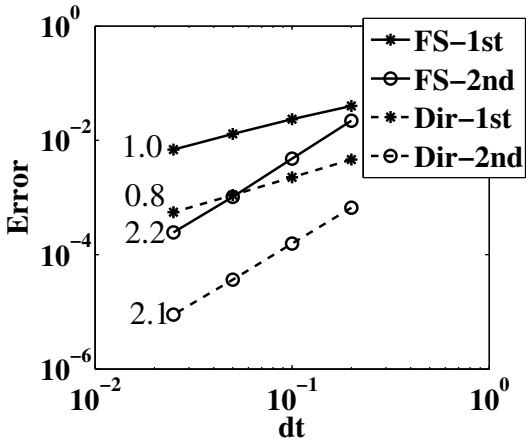


Figure D.1: Taylor-Green vortex problem in 2D: time convergence for the fractional step methods of first and second-order, and the direct linear solver.

- direct solution with Gauss elimination of the original saddle-point;
- iterative solution with GMRES of the original saddle-point linear system, using as pre-conditioner the incomplete LU (ILU) factorization with tolerance ϵ_{ILU} of the original matrix;
- the solution of the two steps of the first-order projection scheme with GMRES, with incomplete LU factorization with zero fill-in (ILU0) as pre-conditioner (note that this kind of pre-conditioner can not be used for the saddle-point linear system because of the zero block on the diagonal);
- the solution of the matrix generated by the Augmented Lagrangian (AL) proposed by Nguyen et al. (2011c).

For the Augmented Lagrangian (AL) method, a relaxation parameter $\beta = 10^{-5}$ for the pressure is used. This value is chosen to guarantee the convergence of the pressure problem. The linearization of the convective term is performed using the steady-state solution of the problem, and a null initial vector is set. The time step is set to $\Delta t = 10^{-2}$.

Table D.2 summarizes the results of the numerical test. The performance of GMRES on the saddle-point linear system strongly depends on the tolerance of the ILU

Table D.2: NACA 0012 airfoil at Re= 1000 and $\alpha = 2°$, mesh $p = 4$ and $\Delta t = 0.01$: comparison of the linear system solution using a direct solver on the original saddle point system and pre-conditioned GMRES on the original saddle point system, the linear system generated by the Augmented Lagrangian, and the two linear systems of the first-order fractional step method. The $\times$ symbol indicates a singular pre-conditioner.

	Dir. sol.	**GMRES**					
		Saddle point system			**AL**	**1st step**	**2nd step**
ϵ_{ILU}		10^{-1}	10^{-2}	10^{-3}	ILU0	ILU0	ILU0
iterations		$\times$	802	55	567	16	129
time prec. [s]		0.77	13	58	0.03	0.03	0.005
time solve [s]		$\times$	120	17	32	0.16	1.1
time tot [s]	4.3	$\times$	133	75	32	0.19	1.1

pre-conditioner. A singular pre-conditioner can be obtained, in this case using a tolerance for the incomplete LU factorization $\epsilon_{ILU} = 10^{-1}$. As a consequence, it is difficult to provide a robust behavior. In all the cases, the solution requires a large computing time compared to the solution of the two steps of the fractional step scheme. The matrix generated by the AL method is ill conditioned, thus the runtime for the solution with GMRES is also large. The projection scheme provides by far the fastest and most robust solution.

A similar comparison has been carried out with a 3D problem, that is, the Taylor-Green vortex problem at Re= 1600 in a cube. The setting of the problem is the one proposed in test-case 3.5 of the 1st International Workshop on High-Order CFD Methods at the 50th AIAA Aerospace Sciences Meeting, Nashville, TN, 2012. An extensive study of this problem can be found in van Rees et al. (2011). Considering a periodic square box defined as $-1 < x, y, z < 1$, zero body forces and the initial condition

$$\boldsymbol{u} = \begin{pmatrix} \sin(x\pi)\cos(y\pi)\cos(z\pi) \\ -\cos(x\pi)\sin(y\pi)\cos(z\pi) \\ 0 \end{pmatrix},$$

this flow transitions to turbulence, with the creation of small scales, followed by a decay phase similar to decaying homogeneous turbulence. In Table D.3 is summarized

Table D.3: Taylor-Green vortex problem in 3D at Re= 1600 in a cubic computational mesh of 3^3 hexahedra $p = 4$ and $\Delta t = 10^{-2}$: comparison of the linear system solution using a direct solver on the original saddle point system and pre-conditioned GMRES on the original saddle point system and the two linear systems of the first-order fractional step method. The $\times$ symbol indicates a singular pre-conditioner.

	Dir. sol.	**GMRES**				
		Saddle point system			**1st step**	**2nd step**
ϵ_{ILU}		10^{-1}	10^{-2}	10^{-3}	ILU0	ILU0
iterations		$\times$	816	18	7	23
time prec [s]		1.2	45	59	3.1	0.2
time solve [s]		$\times$	234	4	0.3	0.1
time tot [s]	29	$\times$	279	63	3.4	0.3

the study of the solution of the linear system generated with a computational mesh of 3^3 hexahedra $p = 4$. In this case, the first time step $\Delta t = 10^{-2}$ is considered and the convective term is linearized around the initial condition. Results underline once more the excellent performance of the projection scheme.

In Figure D.2 is shown the iso-surfaces of the z-component of the vorticity, at the initial and final time $T = 20/\pi$ of the simulation. The suggested results for this test case include the computation of the kinetic energy dissipation rate as a function of the simulation time. As an example of preliminary study, the profile of the energy dissipation rate ε is compared in Figure D.3 with the reference results, for different discretizations and approximation degrees. The simulation time and the dissipation rate are non-dimensionalised respect to a characteristic velocity and time, see van Rees et al. (2011) for details. For computational reasons, the discretization used is much coarser than the one suggested in the test case specifications, however the promising results encourage to further study the HDG fractional step methods.

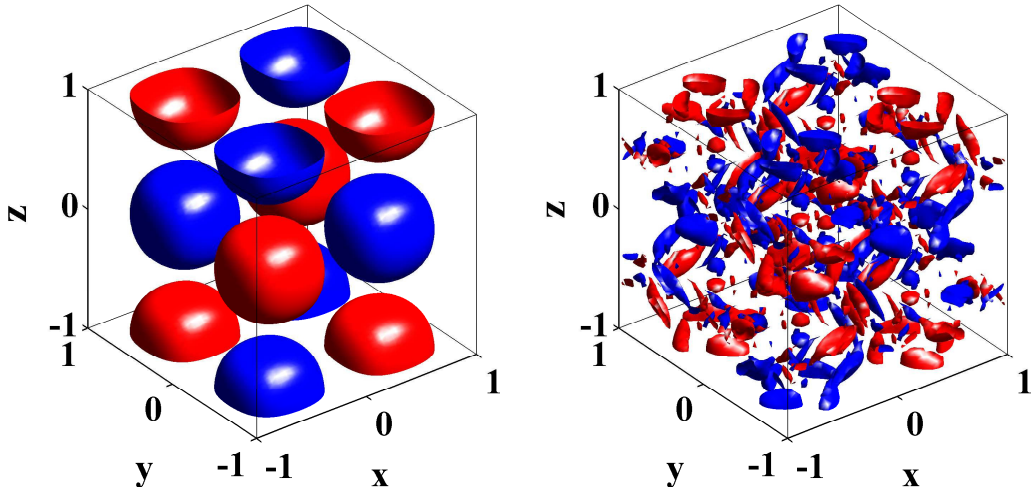


Figure D.2: Taylor-Green vortex problem in 3D at Re= 1600 in a cubic computational mesh of 10^3 hexahedra $p = 4$: iso-surfaces at value 0.25 of the z-component of the vorticity at the initial time (left) and at final time $T = 20/\pi$ (right).

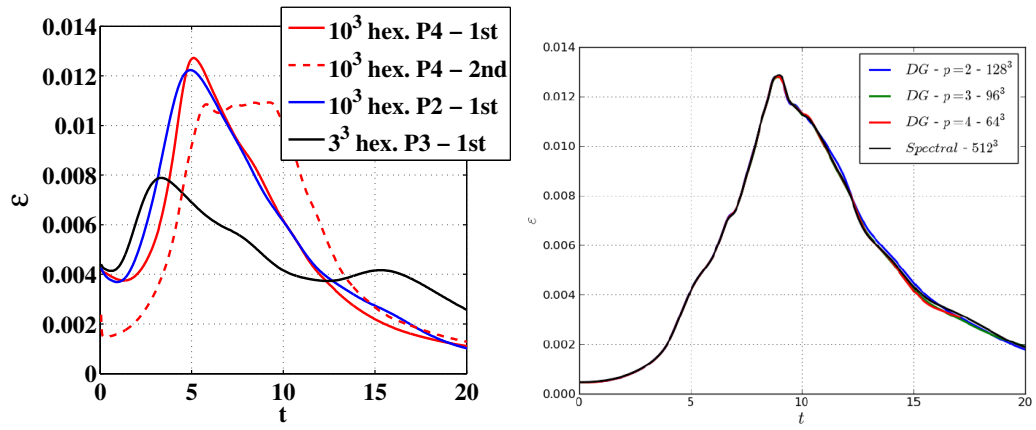


Figure D.3: Taylor-Green vortex problem in 3D at Re= 1600: energy dissipation rate obtained with the HDG projection method (left) and reference values (right).

Appendix E

Discontinuous Galerkin formulation of the Mild Slope equation in time-frequency domain

E.1 Introduction

The MSE, derived by Berkhoff (1972), describes the motion of sea waves over a slow varying bathymetry, and allows to model the refraction and diffraction for deep and shallow water, until a slope of 1/3, see Booij (1983). Starting from the incompressible Navier-Stokes equations, with the hypothesis of non-viscous fluid, small amplitude monochromatic waves and slow varying bottom, the MSE is derived integrating on the vertical structure of the water motion. Details on the derivation of the MSE can be found in Appendix F.

The MSE is a useful tool for evaluating the wave agitation in coastal zone and in harbors. Its wide range of applicability, respect to the bottom depth, allows to simulate the wave propagation from the open sea to the interior of the harbor. Non linear effects such as non-linear interactions between different frequencies, wave breaking, friction with the bottom and sea currents can be included with further development of the MSE (see for example Khellaf and Bouhadef (2004)).

The MSE in frequency domain is

$$\boldsymbol{\nabla} \cdot cc_g \boldsymbol{\nabla} u_T + k^2 cc_g u_T = 0, \tag{E.1}$$

where $u_T(x, y)$ is the surface elevation, $k(x, y)$ is the wave number, $c(x, y)$ is the phase velocity and $c_g(x, y)$ is the group velocity. Equation (E.1) models the steady state condition of a traveling wave with an angular frequency ω given by the dispersion relation Equation (F.33), introduced in Appendix F. Equation (F.33) relates the wavenumber with the bottom depth and the frequency, modeling therefore the effect of the bathymetry on the wave propagation, that is, the *refraction*. Modifying the derivation as detailed by Smith and Sprinks (1975), see also Appendix F, it is possible to obtain an equation that describes the temporal evolution of the wave train. The MSE in time-frequency domain is

$$\frac{\partial^2 u_T}{\partial t^2} - \boldsymbol{\nabla} \cdot (cc_g \boldsymbol{\nabla} u_T) + (\omega^2 - k^2 cc_g) u_T = 0, \tag{E.2}$$

having obviously a steady state condition that verifies (E.1).

Another form of the MSE in time-frequency formulation is often found in literature, see for example Oliveira (2001) and Bokaris and Anastasiou (2003). This alternative form, called the *harmonic form*, is derived imposing that the steady state condition has the harmonic form

$$u_T(x, y, t) = \phi(x, y) \exp(I\omega t)$$

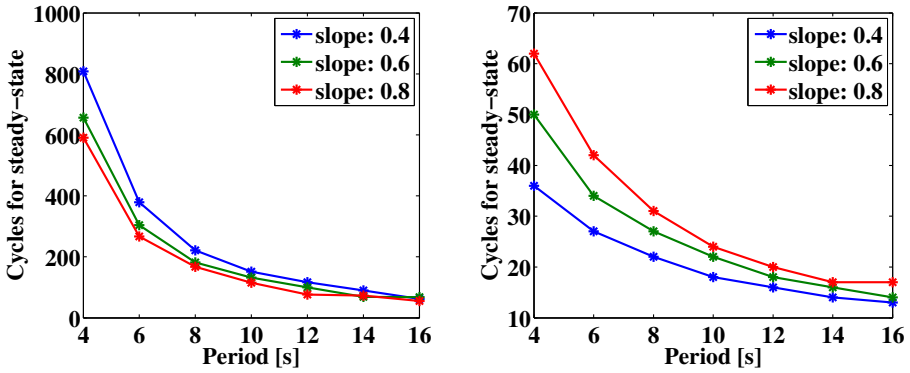
so that

$$\frac{\partial^2 u_T}{\partial t^2} = -\omega^2 u_T. \tag{E.3}$$

Introducing (E.3) in (E.2), the harmonic form of the Mild Slope equation in time-frequency domain is

$$\frac{c_g}{c} \frac{\partial^2 u_T}{\partial t^2} - \boldsymbol{\nabla} \cdot (cc_g \boldsymbol{\nabla} u_T) = 0. \tag{E.4}$$

The steady state of (E.2) and (E.4) is obviously the same. In most of the cases, the steady state configuration of the wave agitation is what drives engineers in harbor



(a) Original form of the MSE (b) Harmonic form of the MSE

Figure E.1: Cycles to convergence to a steady-state solution for the MSE in original form (left) and in harmonic form (right), for a 1D shoaling problem with three different values of the bottom slope.

design. Figure E.1 shows the number of cycles to reach a steady-state solution as a function of the period of the wave, for the original form of the MSE (E.2) and the harmonic form (E.4). The problem considered is a unidirectional wave motion in the x-direction, for $x = [0, 800]$m, with linear bottom slope (linear shoaling problem, see Dingemans (1997)). Three decreasing bottom depths are considered, starting with a depth of 10m and with slope $0.4, 0.6$ and 0.8, and period of the wave varying from 4s to 16s. The original form has a transient state that is much longer than the harmonic form, needing a number of cycle to converge that is one order of magnitude larger. For this reason the harmonic form is considered in the following section for modeling the wave propagation in coastal areas.

E.2 Problem statement

The boundary conditions introduced in Section 2.2 are considered here. The Sommerfeld radiation condition in time domain is

$$\lim_{r\to\infty} \sqrt{r}(\boldsymbol{n}\cdot\boldsymbol{\nabla}u + \frac{1}{c}\frac{\partial u}{\partial t}) = 0, \tag{E.5}$$

where r is the radial direction. Condition (E.5) is applied to the *reflected* wave u, hence it requires to divide the solution of (E.4) into two contributions: the *reflected* wave u (unknown) and the incident wave u_0 (known),

$$u_T = u_0 + u, \tag{E.6}$$

where the incident wave is $u_0 = \cos(\omega t - k_0(x\cos\theta_0 + y\sin\theta_0))$, with k_0 its wave number and θ_0 its direction. A first order non-reflecting boundary condition is used to discretize Equation (E.5) on the artificial boundary, that is

$$\boldsymbol{n}\cdot\boldsymbol{\nabla}u + \frac{1}{c}\frac{\partial u}{\partial t} + \frac{1}{2R} = 0, \qquad \text{on } \Gamma_{\texttt{NRB}},$$

being R the radius of the artificial boundary.

Other boundary conditions are the one modeling physical obstacles, like beaches, dikes and breakwaters, and the one modeling symmetric boundaries. Physical boundaries are threated as partial reflecting boundaries, $\Gamma_{\texttt{R}}$. The amount of energy that the boundary absorbs when the wave impacts on it is called *absorption parameter*, $\alpha \in [0,1]$, and takes zero value on totally reflecting boundaries and unitary value on a total absorbing boundaries. Thus, the partial absorbing boundary condition is

$$\boldsymbol{n}\cdot\boldsymbol{\nabla}u_T + \frac{\alpha}{c}\frac{\partial u_T}{\partial t} = 0, \qquad \text{on } \Gamma_{\texttt{R}}.$$

Finally, symmetric boundary conditions prescribe zero derivative of the solution orthogonal to the boundary, that is

$$\boldsymbol{n}\cdot\boldsymbol{\nabla}u = 0, \text{ on } \Gamma_{\texttt{S}}.$$

Making use of the potential decomposition (E.6), the problem statement for the

harmonic form of the MSE in time-frequency domain becomes

$$\begin{aligned}
\frac{c_g}{c}\frac{\partial^2 u}{\partial t^2} - \boldsymbol{\nabla}\cdot(cc_g\boldsymbol{\nabla}u) &= -B(u_0), && \text{in } \Omega \\
\boldsymbol{n}\cdot\boldsymbol{\nabla}u + \frac{\alpha}{c}\frac{\partial u}{\partial t} &= -(\boldsymbol{n}\cdot\boldsymbol{\nabla}u_0 + \frac{\alpha}{c}\frac{\partial u_0}{\partial t}), && \text{on } \Gamma_{\mathtt{R}} \qquad \text{(E.7)} \\
\boldsymbol{n}\cdot\boldsymbol{\nabla}u + \frac{1}{c}\frac{\partial u}{\partial t} + \frac{1}{2R} &= 0, && \text{on } \Gamma_{\mathtt{NRB}} \qquad \text{(E.8)} \\
\boldsymbol{n}\cdot\boldsymbol{\nabla}u &= 0, && \text{on } \Gamma_{\mathtt{S}}, \qquad \text{(E.9)}
\end{aligned}$$

where the Berkhoff operator is

$$B(\psi) = \frac{c_g}{c}\frac{\partial^2\psi}{\partial t^2} - \boldsymbol{\nabla}\cdot(cc_g\boldsymbol{\nabla}\psi).$$

E.2.1 DG formulation of the MSE in time-frequency domain

The advantage of using the time-frequency formulation respect to the frequency formulation described in Section 2.2 is the important save in memory obtained using an explicit time integration scheme with a DG spatial discretization. In fact, the discontinuous character of the solution and the explicit time integration allow to advance in time the integration separately for each element. No global system of equations are to be solved for the degrees of freedom of the whole computational mesh. The time advancing reduces then to an element-by-element solution of a small linear system represented by the elemental mass matrix.

The DG method with *upwind* scheme for the numerical fluxes is used to model Equation (E.4), see for example Hesthaven and Warburton (2002). Introducing some auxiliary variables

$$r := \frac{\partial u}{\partial t}, \quad q_j := \frac{\partial u}{\partial x_j},$$

where the index $j = 1, 2$ refers to the coordinates x and y, the problem is rewritten as a system of 1st order hyperbolic equations

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_j}{\partial x_j} = \boldsymbol{B}(u_0), \qquad \text{(E.10)}$$

where repeated indexes sum over. The variables are

$$\boldsymbol{U} = \begin{Bmatrix} c_g/c\ r \\ q_1 \\ q_2 \\ u \end{Bmatrix} := \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix};$$

$$\boldsymbol{F}_1 = \begin{Bmatrix} -cc_g q_1 \\ -r \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -cc_g U_2 \\ -U_1 \\ 0 \\ 0 \end{Bmatrix}; \quad \boldsymbol{F}_2 = \begin{Bmatrix} -cc_g q_2 \\ 0 \\ -r \\ 0 \end{Bmatrix} = \begin{Bmatrix} -cc_g U_3 \\ 0 \\ -U_1 \\ 0 \end{Bmatrix};$$

$$\boldsymbol{B}(u_0) = \begin{Bmatrix} -B(u_0) \\ 0 \\ 0 \\ -r \end{Bmatrix} := \begin{Bmatrix} -B(u_0) \\ 0 \\ 0 \\ -U_1 \end{Bmatrix};$$

Multiplying (E.10) by a test vector $\boldsymbol{W} \in [\widehat{\mathcal{V}}_h]^4$ and integrating in one element Ω_i gives

$$(\boldsymbol{W}, \frac{\partial \boldsymbol{U}_i}{\partial t})_{\Omega_i} + (\boldsymbol{W}, \frac{\partial \boldsymbol{F}_j(\boldsymbol{U}_i)}{\partial x_j})_{\Omega_i} = (\boldsymbol{W}, \boldsymbol{B})_{\Omega_i},$$

where $\boldsymbol{U}_i$ denotes the restriction of $\boldsymbol{U}$ to the element Ω_i. Integrating by parts results

$$(\boldsymbol{W}, \frac{\partial \boldsymbol{U}_i}{\partial t})_{\Omega_i} - (\frac{\partial \boldsymbol{W}}{\partial x_j}, \boldsymbol{F}_j(\boldsymbol{U}_i))_{\Omega_i} + \langle \boldsymbol{W}, \boldsymbol{F}_j(\boldsymbol{U}_i) n_j \rangle_{\partial\Omega_i} = (\boldsymbol{W}, \boldsymbol{B})_{\Omega_i}. \tag{E.11}$$

To impose continuity in weak form between the elements, the flux in Equation (E.11) is replaced by a numerical flux $\tilde{\boldsymbol{F}}_n(\boldsymbol{U}_i, \boldsymbol{U}_i^{out})$ that depends on the solution in the two neighboring elements sharing the face where the numerical flux is defined, obtaining

$$(\boldsymbol{W}, \frac{\partial \boldsymbol{U}_i}{\partial t})_{\Omega_i} - (\frac{\partial \boldsymbol{W}}{\partial x_j}, \boldsymbol{F}_j(\boldsymbol{U}_i))_{\Omega_i} + \langle \boldsymbol{W}, \tilde{\boldsymbol{F}}_n(\boldsymbol{U}_i, \boldsymbol{U}_i^{out}) \rangle_{\partial\Omega_i} = (\boldsymbol{W}, \boldsymbol{B})_{\Omega_i}, \tag{E.12}$$

where $\boldsymbol{U}_i^{out}$ is the solution evaluated at the neighboring element. The choice of the numerical fluxes is crucial for the stability of the method. A natural choice for linear

hyperbolic problems is the flux splitting technique described by Donea and Huerta (2003), which corresponds to an upwind approximation. The flux is first decomposed in the sum of two contribution, the incoming flux (-) and the outgoing flux (+), that is

$$\boldsymbol{F}_n := \boldsymbol{F}_j n_j = \boldsymbol{F}_n^- + \boldsymbol{F}_n^+,$$

then, the numerical flux is chosen as

$$\tilde{\boldsymbol{F}}_n(\boldsymbol{U}, \boldsymbol{U}^{out}) = \boldsymbol{F}_n^-(\boldsymbol{U}) + \boldsymbol{F}_n^+(\boldsymbol{U}^{out}),$$

that is, the outgoing flux is calculated using the unknowns in the current element Ω_i, while the incoming flux is evaluated using the unknowns in the neighboring element.

Due to the linearity of the PDE, the flux can be written as

$$\boldsymbol{F}_n = \mathbf{A_n}\boldsymbol{U},$$

with the Jacobian matrix $\mathbf{A_n} = \frac{\partial \boldsymbol{F}_n}{\partial \boldsymbol{U}}$. The matrix $\mathbf{A_n}$ can be re-written using the spectral decomposition

$$\mathbf{A_n} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1},$$

being $\mathbf{V}$ is the matrix of eigenvectors of $\mathbf{A_n}$ and $\mathbf{D}$ is the diagonal matrix containing the eigenvalues of $\mathbf{A_n}$. Thus, defining the two matrices

$$\mathbf{A_n^+} := \frac{1}{2}(\mathbf{A_n} + |\mathbf{A_n}|), \quad \mathbf{A_n^-} := \frac{1}{2}(\mathbf{A_n} - |\mathbf{A_n}|),$$

where $|\mathbf{A_n}| = \mathbf{V}|\mathbf{D}|\mathbf{V}^{-1}$, results $\mathbf{A_n} = \mathbf{A_n^-} + \mathbf{A_n^+}$. The incoming and outgoing fluxes can be written as

$$\boldsymbol{F}_n^- = \mathbf{A_n^-}\boldsymbol{U}, \quad \boldsymbol{F}_n^+ = \mathbf{A_n^+}\boldsymbol{U},$$

so the numerical flux is

$$\tilde{\boldsymbol{F}}_n(\boldsymbol{U}, \boldsymbol{U}^{out}) = \mathbf{A_n^+}\boldsymbol{U} + \mathbf{A_n^-}\boldsymbol{U}^{out}.$$

Substituting the expressions for $\boldsymbol{F}$, the flux is

$$\boldsymbol{F}_n = \boldsymbol{F}_1 n_1 + \boldsymbol{F}_2 n_2 = \begin{Bmatrix} -cc_g(U_2 n_1 + U_3 n_2) \\ -n_1 U_1 \\ -n_2 U_1 \\ 0 \end{Bmatrix},$$

giving the matrix

$$\mathbf{A_n} = \begin{bmatrix} 0 & -cc_g n_1 & -cc_g n_2 & 0 \\ -n_1 & 0 & 0 & 0 \\ -n_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The spectral decomposition is

$$\mathbf{A_n} = \mathbf{VDV}^{-1} =$$

$$\begin{bmatrix} 0 & 0 & -\sqrt{cc_g}/n_2 & \sqrt{cc_g}/n_2 \\ 0 & -n_2/n_1 & n_1/n_2 & n_1/n_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{cc_g} & 0 \\ 0 & 0 & 0 & -\sqrt{cc_g} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & -n_2 n_1 & n_1^2 & 0 \\ -\frac{1}{2} n_2/\sqrt{cc_g} & \frac{1}{2} n_1 n_2 & \frac{1}{2} n_2^2 & 0 \\ \frac{1}{2} n_2/\sqrt{cc_g} & \frac{1}{2} n_1 n_2 & \frac{1}{2} n_2^2 & 0 \end{bmatrix},$$

leading to the matrices

$$|\mathbf{A_n}| = \begin{bmatrix} \sqrt{cc_g} & 0 & 0 & 0 \\ 0 & n_1^2\sqrt{cc_g} & n_1 n_2\sqrt{cc_g} & 0 \\ 0 & n_1 n_2\sqrt{cc_g} & n_2^2\sqrt{cc_g} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{A_n^-} = -\frac{1}{2} \begin{bmatrix} \sqrt{cc_g} & cc_g n_1 & cc_g n_2 & 0 \\ n_1 & n_1^2\sqrt{cc_g} & n_1 n_2\sqrt{cc_g} & 0 \\ n_2 & n_1 n_2\sqrt{cc_g} & n_2^2\sqrt{cc_g} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{A}_{\mathbf{n}}^{+} = \frac{1}{2}\begin{bmatrix} \sqrt{cc_g} & -cc_g n_1 & -cc_g n_2 & 0 \\ -n_1 & n_1^2\sqrt{cc_g} & n_1 n_2\sqrt{cc_g} & 0 \\ -n_2 & n_1 n_2\sqrt{cc_g} & n_2^2\sqrt{cc_g} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and resulting in the numerical flux

$$\tilde{\boldsymbol{F}}_n(\boldsymbol{U}, \boldsymbol{U}^{out}) = \frac{1}{2}\begin{Bmatrix} c\sqrt{cc_g}\left[(U_1 - U_1^{out}) - \sqrt{cc_g}\left(n_1(U_2 + U_2^{out}) + n_2(U_3 + U_3^{out})\right)\right] \\ n_1\left[-(U_1 + U_1^{out}) + \sqrt{cc_g}\left(n_1(U_2 - U_2^{out}) + n_2(U_3 - U_3^{out})\right)\right] \\ n_2\left[-(U_1 + U_1^{out}) + \sqrt{cc_g}\left(n_1(U_2 - U_2^{out}) + n_2(U_3 - U_3^{out})\right)\right] \\ 0 \end{Bmatrix}.$$

Calling now

$$\Delta = n_1 U_2 + n_2 U_3,$$

the numerical flux can be rewritten as

$$\tilde{\boldsymbol{F}}_n(\boldsymbol{U}, \boldsymbol{U}^{out}) = \frac{1}{2}\begin{Bmatrix} \sqrt{cc_g}\left[(U_1 - U_1^{out}) - \sqrt{cc_g}\left(\Delta^{out} + \Delta\right)\right] \\ n_1\left[-(U_1 + U_1^{out}) + \sqrt{cc_g}\left(\Delta - \Delta^{out}\right)\right] \\ n_2\left[-(U_1 + U_1^{out}) + \sqrt{cc_g}\left(\Delta - \Delta^{out}\right)\right] \\ 0 \end{Bmatrix}.$$

The numerical flux in each face of a generic element Ω_i depends on the values of U_1^{out} and Δ^{out} in the neighboring element. This is crucial for the definition of the flux in the boundary elements, as explained in the next section.

The discretization of (E.12) in each element, see Appendix B, leads to a system of ordinary differential equations (ODEs)

$$\mathbf{M}\frac{d\mathbb{U}_i}{dt} + \mathbf{R}(\mathbb{U}_i, \mathbb{U}_i^{out}) = \mathbf{B}, \tag{E.13}$$

where $\mathbb{U}_i$ is the vector of nodal values, $\mathbf{M}$ is the elemental mass matrix and $\mathbf{R}(\mathbb{U}_i, \mathbb{U}_i^{out})$ is a residual vector containing the contribution of the neighboring element through the numerical fluxes.

The ODE system (E.13) is advanced in time using a standard explicit fourth-order Runge-Kutta scheme, see for instance Donea and Huerta (2003). The stability condition requires a time step

$$\Delta t \leq \tilde{C}\frac{h_{min}}{cp^2} \tag{E.14}$$

where $\tilde{C}$ is a constant, h_{min} is the minimum element size in the mesh and c is the phase velocity.

The time marching process is stopped when the relative error in the solution between two consecutive cycles is small enough.

E.2.2 Boundary Conditions

Rankine-Ugoniot jump conditions

Each boundary condition provides a relation that must be fulfilled on the exterior face of each boundary element. Thus, the boundary condition in the boundary element provides one of the two relations needed to compute the numerical flux. The second relation that allows to compute U_1^{out} and Δ^{out} is given by the Rankine-Ugoniot jump conditions, see LeVeque (1992), that is

$$[\![\boldsymbol{F}_n]\!] = \lambda_e [\![\boldsymbol{U}]\!], \tag{E.15}$$

where λ_e are the eigenvalues of the matrix $\mathbf{A_n}$,

$$\lambda_1 = \lambda_2 = 0, \quad \lambda_3 = -\lambda_4 = \sqrt{cc_g}.$$

Equation E.15 specialized for a boundary face is

$$[\![\boldsymbol{F}_n]\!] = -\sqrt{cc_g}[\![\boldsymbol{U}]\!],$$

so the Rankine-Ugoniot jump conditions give

$$\begin{aligned} cc_g[\![\Delta]\!] &= \sqrt{cc_g}[\![U_1]\!], \\ n_1[\![U_1]\!] &= \sqrt{cc_g}[\![U_2]\!], \\ n_2[\![U_1]\!] &= \sqrt{cc_g}[\![U_3]\!], \\ 0 &= \sqrt{cc_g}[\![U_4]\!]. \end{aligned}$$

Partial absorbing boundary condition

With the notation introduced in the previous section, condition (E.7) can be written as

$$\Delta^{out} + \frac{\alpha}{c} U_1^{out} = g,$$

where the incident potential information is

$$g = -(\frac{\partial u_0}{\partial \boldsymbol{n}} + \frac{\alpha}{c}\frac{\partial u_0}{\partial t}).$$

Using the Rankine-Ugoniot jump condition $\sqrt{cc_g}[\![\Delta]\!] = [\![U_1]\!]$, the two relations become

$$\begin{aligned} U_1^{out} &= \frac{U_1 + \sqrt{cc_g}(g - \Delta)}{1 + \alpha\sqrt{c_g/c}}, \\ \Delta^{out} &= \frac{g + \alpha/c(\sqrt{cc_g}\Delta - U_1)}{1 + \alpha\sqrt{c_g/c}}, \end{aligned}$$

that are the relations needed to compute the numerical flux $\tilde{\boldsymbol{F}}_n$ on the exterior face of the boundary element.

Radiation boundary condition

The radiation boundary condition (E.8) can be rewritten as

$$\Delta^{out} + \frac{1}{c} U_1^{out} + \frac{1}{2R} U_4^{out} = 0,$$

that, with the Rankine-Ugoniot jump conditions $\sqrt{cc_g}[\![\Delta]\!] = [\![U_1]\!]$ and $[\![U_4]\!] = 0$, gives

$$U_1^{out} = \frac{U_1 - \sqrt{cc_g}(\Delta - U_4/2R)}{1 + \sqrt{c_g/c}},$$
$$\Delta^{out} = \frac{1/c\left(\sqrt{cc_g}(\Delta + U_4/2R) - U_1\right)}{1 + \sqrt{c_g/c}}.$$

Symmetry boundary condition

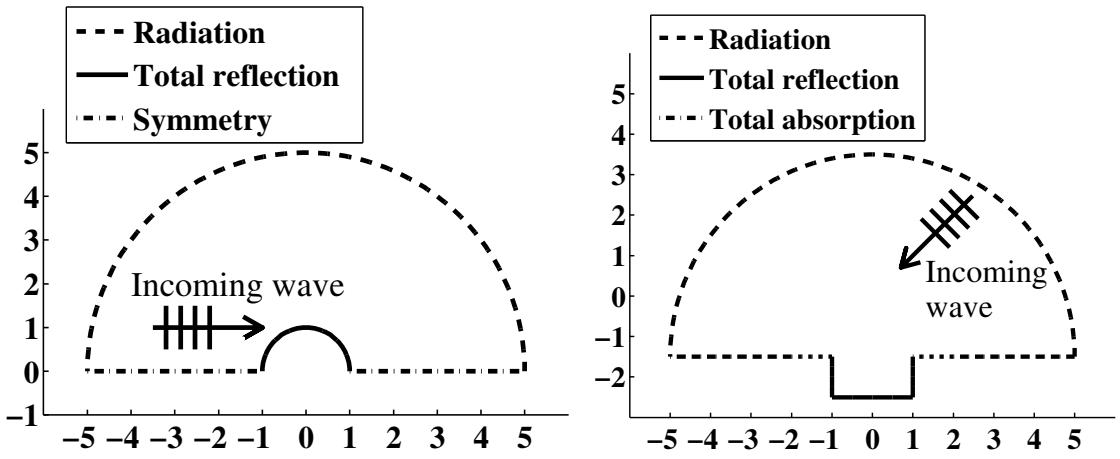
The symmetry condition (E.9) can be rewritten simply as $\Delta^{out} = 0$. Using again the Rankine-Ugoniot jump conditions, the two relations are

$$U_1^{out} = U_1 - \sqrt{cc_g}\Delta \quad \text{and} \quad \Delta^{out} = \Delta.$$

E.2.3 Numerical experiments

A set of simulations has been carried out to estimate the computational cost of solving the wave propagation in a harbor using equation (E.4) discretized with the DG scheme introduced in the last section. Two different geometries have been considered to study the effect of the scattered field on the number of cycles needed to reach the equilibrium state. The first geometry models a circular scatterer similar to the one studied in Section 2.2.3. The surface of the scatterer is considered totally reflecting. Due to the symmetry of the problem, only half of the domain is simulated as shown in Figure E.2(a). The second problem regards the scattering of a plane wave in rectangular cavity, similar to the problem of Section 2.2.3, and its geometry is shown in Figure E.2(b). In this case, the rectangular region is totally reflecting while the rest of the reflecting boundary is totally absorbing. In both problems, the totally reflecting boundaries are considered as the areas of interest of the problem. For simplicity reason, a constant bottom depth is considered in both geometries and no PML is considered.

The problem is solved for three wavenumbers of the incident wave, $k = 3, 6, 9$, using triangular meshes. Three element sizes are considered, $h = 0.5, 1, 2$, and the



(a) Scattering by a circular obstacle (b) Scattering by a rectangular cavity

Figure E.2: Setting of the problem for the circular scattering and the rectangular cavity.

polynomial degree is $p = 1 \cdots 10$. In Figure E.3 are depicted the solutions for a wave number $k = 9$, corresponding to a wavelength of about 0.7. The time to compute one single period of the incident wave is evaluated for the different discretizations. Figure E.4 shows the runtime for simulating one single period of the incident wave with $k = 9$ for the two geometries. It is important to note that the runtime only depends on the wave resolution (i.e. the number of nodes per wavelength, see Section 2.2) and not on the discretization and the problem. This means that meshes with different element size and different p but the same number of nodes per wavelength lead to the same runtime for one period.

In Figure E.5 is shown the convergence to the steady state solution. This result highlights the dependence of the convergence history from the problem, in particular the importance of the interactions of the scattered field, and the frequency of the problem. In fact, for the rectangular cavity where the scattering field is much more complex, the convergence appears more irregular.

The estimation of the runtime for a real case of harbor simulation is extrapolated from the curves of the runtime for one period of the incident wave for the two examples considered. Referring to Figure E.4 and considering to increase the wavenumber,

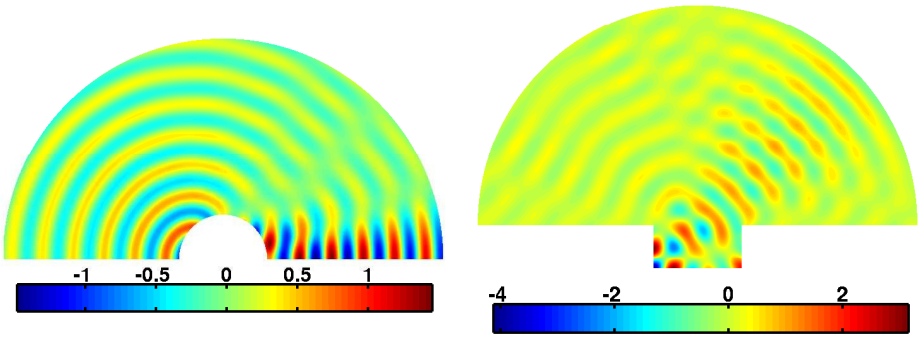


Figure E.3: Solution for $k = 9$ for the circular scattering (left) and the rectangular cavity (right).

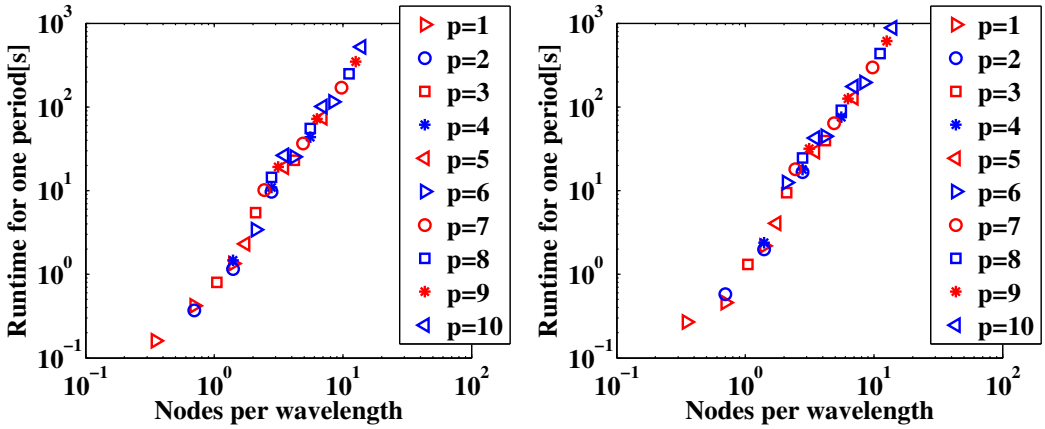


Figure E.4: Runtime for one period of the incident wave with $k = 9$ for the scattering circle geometry (left) and the rectangular cavity (right).

the result would be a translation of the points to the left (for the decrease of the wavelenght), and also downward (for the increase of the time step, i.e. $\Delta t \propto k$, see Equation (E.14) considering $c \propto 1/k$). Following this rationale, it is possible to deduce a plot similar to the one of Figure E.4 but considering a number of waves in the computational domain realistic for a harbor simulation. In fact, if in the numerical tests depicted in Figure E.2(a) and Figure E.2(b) only about 7 waves are present in the domain, in a realistic large size harbor simulation at least 100 wave are to be considered, see also the numerical example of the Barcelona harbor in Section 2.2.3. In Figure E.6, the resulting interpolated curves corresponding to $k = 100\ 9/7$ are shown. The approximated time to compute one period of the incident wave is

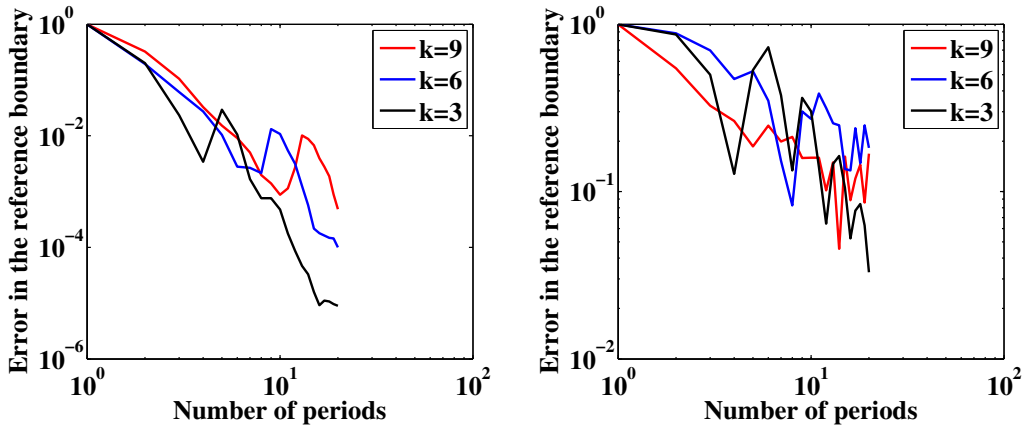


Figure E.5: Convergence to the steady state solution for the scattering circle geometry (left) and for the rectangular cavity (right), for $k = 3, 6, 9$.

also displayed considering a wave resolution of 10 nodes per wavelength. Using both examples, the estimated time to solve one single period of harbor simulation is of the order 10^3 seconds. It is expectable that, due to the complexity of the solution in a large harbor, hundreds of cycles are needed to obtain a steady state solution. Moreover, the need to capture small geometrical features in the interior of the harbor often require small element sizes with consequent decrease of the allowable time step. Thus, the computational effort results prohibitive compared to the runtimes obtained for solving the Barcelona harbor using the Mild Slope equation in frequency formulation.

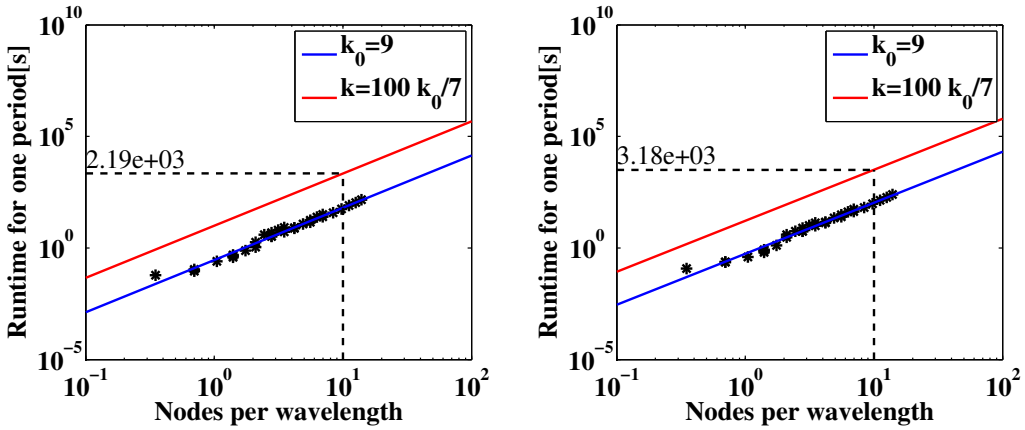


Figure E.6: Estimated time to compute one single period of the incident wave for the case of 100 waves in the domain. In the left figure, the estimation is done considering the problem of the circular scattering, in the right one considering the rectangular cavity

Appendix F

Derivation of the Mild Slope equation*

*** written with David Modesto**

F.1 Physical problem

The ocean swell, due to the friction of the wind and the resulting exchange of energy with the free surface of the water, has been objective of study and simulation for years (see for example Berkhoff (1972) and Bettess and Zienkiewicz (1977)). Reproducing the energy transport and modification by means of several physical fenomena is crucial when simulating the variation of the free surface of the sea and its effects on multiple engineering applications. This transport depends largely on two types of physical contours: those in the domain of propagation, such as dikes, breakwaters, walls or beaches, plus the seabed itself. The two contours influence the propagation of the waves differently but with common consequences: both are capable of causing a change of direction of the wave advance and also a variation of its height. Moreover, the bathymetry affects the propagation speed and the wavelength. Others effects are also present, such as breaking waves, sea currents and bottom friction.

Two dominant effects occur when a wave, as it propagates, hits a boundary. If the

contour reflects, completely or partially, the energy of the incident wave, *reflection* occurs, that is, a change of direction of the propagation which depends on the angle of incidence. Moreover, every object has associated an adjacent area where the wave has no effect. With no propagation in these areas, the energy is low in comparison with the points of incidence, for this reason such areas are known by the name of *shadow areas*. If the dimension of the obstacle is much greater than the wavelength, the shadow area is almost total and the only visible effect is the reflection. On the contrary if the wave is very long in comparison with the size of the object, the shadow area is smaller in comparison to the incidence area. It is in this situation where high energy gradients between both areas cause a propagation perpendicular to the advancing wavefront. This effect is called *diffraction* of the wave.

On the other hand, as the wave propagates, the oscillatory motion of the particles forming the free surface tends to generate another motion in the fluid. The direction of this motion is perpendicular to the plane of propagation and its trajectory is elliptic (a schematic representation of this movement can be seen in Figure F.1). This oscillatory motion "connects" the bottom with the surface of the sea. Thus, when the water is not deep enough, this current rubs with the seabed, exchanging energy and decreasing the speed of the wave train. The more shallow the water is, the greater is the effect produced by the friction with the seabed. However, when the water is deep enough, the movement of the free surface can not affect the fluid particles closer to the bottom, hence the wave speed is not altered (see in Figure F.1 as the elliptic movement decreases with depth). For this reason, if the bathymetry is not constant, different groups of surface waves are generated at different point of the propagation domain. The physical consequence is that the swell rotates to obtain a direction as parallel as possible to the maximum bottom slope line. This effect is known as the *refraction* of the wave.

Here, the case of wave propagation over a varying bottom depth is considered, to simulate both diffraction and reflection, which are the only effects present under constant bathymetry, and also refraction. In Figure F.2 this model is presented, where $\eta(x, y, t)$ denotes the free surface, $h(x, y)$ is the bottom depth, D the mean bottom

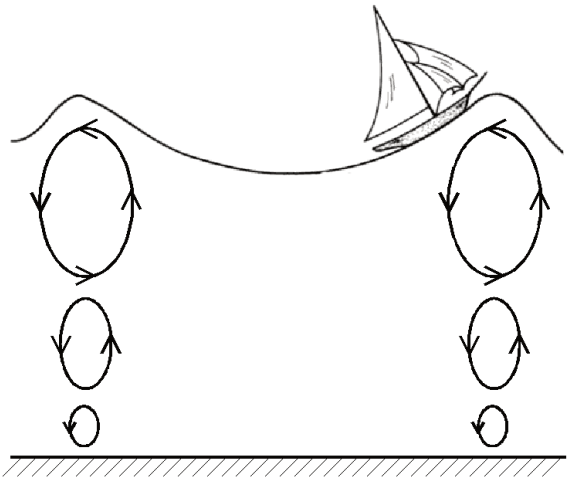


Figure F.1: Elliptic movement perpendicular to the wave propagation direction.

depth, H is the wave height and ℓ is the wavelength. In the following, the obvious condition $\eta(x, y) \leq h(x, y)$ is considered. Moreover, the additional condition on the characteristic dimensions is made,

$$\ell \geq H.$$

Obviously this hypothesis is never a restriction for sea waves.

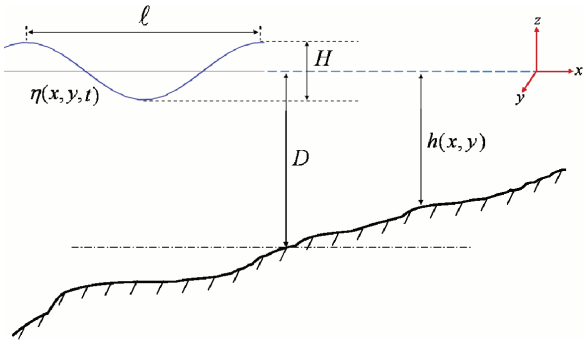


Figure F.2: Wave propagation profile over an uneven bottom.

F.2 Non-linear mathematical formulation

In this section is presented the equations and boundary conditions to model non-linear waves. An ideal, homogeneous, incompressible fluid is considered. Using these

hypothesis and developing the equations of continuity and momentum follows the problem of non-linear waves. The model includes two boundary conditions contained in the spatial variable z, based on the fact that both the seabed and the free surface are always formed by the same particles.

Due to the complexity involved in solving this problem, in Sections F.3 and F.4 the governing equations are linearized and approximated to obtain a two dimensional model.

F.2.1 Governing equations

The continuity and momentum equations for an ideal, homogeneous, incompressible flow are

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u} + \frac{1}{\rho}\boldsymbol{\nabla} p = \boldsymbol{b}, \tag{F.1}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0, \tag{F.2}$$

where $\boldsymbol{b}$ is a vector of body forces and ρ is the density of the fluid. Taking the curl of (F.1) and defining the vorticity $\boldsymbol{\omega} = \boldsymbol{\nabla}\times\boldsymbol{u}$, results

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{\nabla}\times(\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}) + \boldsymbol{\nabla}\times\left(\frac{1}{\rho}\boldsymbol{\nabla} p\right) = \mathbf{0}$$

having considered conservative body forces, that is $\boldsymbol{b} = (0, 0, -g)^T$.

Using now some identities of vector calculus[1], (F.1) is directly reduced to

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{\omega} - \boldsymbol{\omega}\boldsymbol{\nabla}\boldsymbol{u} = \mathbf{0}, \tag{F.3}$$

which is the *vorticity equation* for an ideal, homogeneous, incompressible fluid. Note that (F.3) is a transport equation with convective term $\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{\omega}$ ($\boldsymbol{u}$ is the convection speed) and reaction term $\boldsymbol{\omega}\boldsymbol{\nabla}\boldsymbol{u}$. Thus, imposing [2]

[1] $\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u} = \boldsymbol{\nabla}(\frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}) - \boldsymbol{u}\times\boldsymbol{\omega}$, $\boldsymbol{\nabla}\times(\boldsymbol{u}\times\boldsymbol{\omega}) = -\boldsymbol{\omega}\boldsymbol{\nabla}\cdot\boldsymbol{u} + \boldsymbol{\omega}\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{\omega}$, $\boldsymbol{\nabla}\times\boldsymbol{\nabla}\bullet = \mathbf{0}$ and $\boldsymbol{\nabla}\cdot\boldsymbol{\omega} = 0$.

[2] Since the fluid is inviscid, this is not a restriction.

$$\boldsymbol{\omega}|_{t=t_0} = \mathbf{0} \quad \forall (x,y,z) \in \Omega,$$
$$\boldsymbol{\omega}|_{\partial\Omega} = \mathbf{0} \quad \forall t \in \mathbb{R}^+,$$

results

$$\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u} = \mathbf{0} \quad \forall (x,y,z) \in \Omega,\ \forall t \in \mathbb{R}^+ \quad \Longrightarrow \quad \boldsymbol{u} = \boldsymbol{\nabla}\Phi, \tag{F.4}$$

where Φ is a scalar potential. Thus, (F.2) becomes the Laplace equation when (F.4) is applied. A useful way to write it derives from splitting the Laplace operator in the propagation plane $x-y$ and in the out of plane direction z, that is $\Delta = \Delta_{x,y} + \Delta_z = \Delta_2 + \partial^2/\partial z^2$,

$$\boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\Phi = \Delta\Phi = \Delta_2\Phi + \frac{\partial^2 \Phi}{\partial z^2} = 0, \tag{F.5}$$

which is the first governing equation.

The second governing equation comes from substituting (F.4) in (F.1), that is,

$$\frac{\partial}{\partial t}\left(\boldsymbol{\nabla}\Phi\right) + \boldsymbol{\nabla}\Phi \cdot \boldsymbol{\nabla}(\boldsymbol{\nabla}\Phi) + \frac{1}{\rho}\boldsymbol{\nabla} p = \boldsymbol{b},$$

which is equivalent to

$$\boldsymbol{\nabla}\left(\frac{\partial \Phi}{\partial t} + \frac{1}{2}\boldsymbol{\nabla}\Phi \cdot \boldsymbol{\nabla}\Phi + \frac{p}{\rho} + gz\right) = \mathbf{0},$$

where the identity $\boldsymbol{\nabla}(-gz) = \boldsymbol{b}$ has been used. This equation means that the terms in the parenthesis is a generic function of the time only, $E(t)$, thus

$$\frac{\partial \Phi}{\partial t} + \frac{1}{2}\boldsymbol{\nabla}\Phi \cdot \boldsymbol{\nabla}\Phi + \frac{p}{\rho} + gz = E(t), \tag{F.6}$$

which is the so called *Bernoulli equation*, stating an energy balance in the fluid. Since no dissipation is present in an ideal fluid, $E(t)$ can be considered constant in time. Thus, with no loss of generality, it can be taken $E = p_0/\rho$, where p_0 is a reference

pressure. Re-writing (F.6), results

$$\frac{\partial \Phi}{\partial t} + \frac{1}{2}\boldsymbol{\nabla}\Phi \cdot \boldsymbol{\nabla}\Phi + \frac{p - p_0}{\rho} + gz = 0, \tag{F.7}$$

which is the second governing equation.

F.2.2 Boundary condition in the profile

Once obtained the fundamental Equations (F.5) and (F.7) for an ideal, homogeneous, incompressible fluid, the general problem of wave propagation with free surface $z = \eta(x, y, t)$ and sea bottom $z = -h(x, y)$ is considered. Denoting $F_1(x, y, z, t) = z - \eta = 0$ the spatial description of the free surface and $F_2(x, y, z) = z + h = 0$ the spatial description of the bottom surface (see Figures F.3 and F.4), a particle belonging to one of these surfaces at a given time can not leave the surface, thus

$$\frac{DF_i}{Dt} = 0, \quad i = \{1, 2\}\,. \tag{F.8}$$

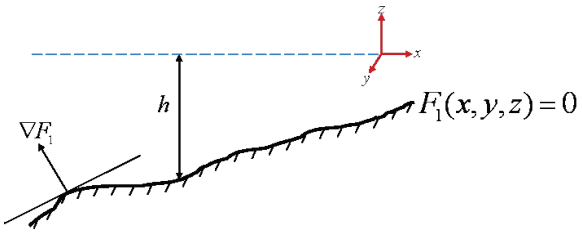


Figure F.3: Sea bottom as material surface.

Rewriting the material derivative in (F.8) results, for the bottom surface ($i = 2$), $\boldsymbol{\nabla}\Phi \cdot \boldsymbol{\nabla}F_2 = 0$, with the obvious meaning that the velocity must be tangent to the bottom surface. Substituting the expression for F_2, the *kinematic bottom condition* results

$$\frac{\partial \Phi}{\partial z} = -\boldsymbol{\nabla}_2\Phi \cdot \boldsymbol{\nabla}_2 h \quad \text{in} \quad z = -h(x, y), \tag{F.9}$$

where the gradient operator in the propagation plane $x - y$, $\boldsymbol{\nabla}_2$, has been introduced.

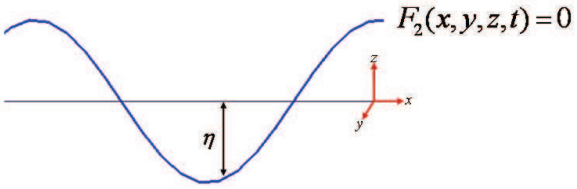


Figure F.4: Free surface as material surface.

For the free surface ($i = 1$), the same argument leads to the *kinematic surface condition*,

$$\frac{\partial \Phi}{\partial z} = \frac{\partial \eta}{\partial t} + \boldsymbol{\nabla}_2 \Phi \cdot \boldsymbol{\nabla}_2 \eta \quad \text{in} \quad z = \eta(x, y, t). \tag{F.10}$$

In order to obtain a boundary condition that only depends on the potential Φ, the second governing Equation (F.7) is evaluated in $z = \eta$ for a reference pressure $p_0 = p_{\text{atm}} = p|_{z=\eta}$, obtaining

$$\eta = -\frac{1}{g}\left(\frac{\partial \Phi}{\partial t} + \frac{1}{2}\boldsymbol{\nabla}\Phi \cdot \boldsymbol{\nabla}\Phi\right). \tag{F.11}$$

This relation allows to compute the free surface once solved the wave problem as a function of the potential Φ. Taking total derivative of (F.11) and using the kinematic surface condition (F.10), the so called *dynamic condition* is obtained,

$$g\frac{\partial \Phi}{\partial z} + \frac{\partial^2 \Phi}{\partial t^2} + \frac{\partial}{\partial t}\left(\boldsymbol{\nabla}\Phi \cdot \boldsymbol{\nabla}\Phi\right) + \boldsymbol{\nabla}\Phi \cdot \left[\boldsymbol{\nabla}\Phi \cdot \boldsymbol{\nabla}(\boldsymbol{\nabla}\Phi)\right] = 0 \quad \text{in} \quad z = \eta(x, y, t), \tag{F.12}$$

which is the desired boundary condition only depending on Φ.

F.2.3 Non-linear wave problem

Summarizing the previous development, the non-linear wave problem in the z variable is defined by the Laplace equation (F.5), the kinematic condition (F.9) for the sea bottom and the dynamic condition (F.12) for the surface. Denoting $\Omega_z = \{(x, y, z) \in \mathbb{R}^3 \mid -h(x, y) < z < \eta(x, y, t)\}$ the domain in the vertical profile, the wave

propagation problem over an uneven bottom is

$$\left.\begin{aligned} \Delta_2\Phi + \frac{\partial^2\Phi}{\partial z^2} &= 0, && \text{in } \Omega_z, \\ g\frac{\partial\Phi}{\partial z} + \frac{\partial^2\Phi}{\partial t^2} + \frac{\partial}{\partial t}\left(\boldsymbol{\nabla}\Phi\cdot\boldsymbol{\nabla}\Phi\right) & \\ +\boldsymbol{\nabla}\Phi\cdot[\boldsymbol{\nabla}\Phi\cdot\boldsymbol{\nabla}(\boldsymbol{\nabla}\Phi),] &= 0, && \text{in } z=\eta(x,y,t), \\ \frac{\partial\Phi}{\partial z} &= -\boldsymbol{\nabla}_2\Phi\cdot\boldsymbol{\nabla}_2 h, && \text{in } z=-h(x,y). \end{aligned}\right\} \tag{F.13}$$

F.3 Linearization

In this section is presented the linearization of problem (F.13). The linearization process involves Equation (F.12), but also the definition of the computational domain Ω_z, which depends on the free surface η, unknown of the problem. Hence, additional restrictions are needed on the shape of the wave, depending on the depth of the sea bottom.

These restriction are deduced from a particular case of one-directional wave propagation over horizontal bottom.

F.3.1 Additional restrictions. *Long-crested linear waves*

Problem (F.13) is restricted to the direction x, and Equation (F.12) is split in the two equivalent Equations (F.10) and (F.11). For constant bottom depth h, the linearization of the equation gives the two dimensional wave propagation problem in the plane $z-x$ (*long-crested linear waves*), ie

$$\frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial z^2} = 0 \qquad \text{in } -h < z < \eta(x,t) \qquad \text{(Laplace)}, \tag{F.14}$$

$$\frac{\partial\Phi}{\partial t} = -g\eta \qquad \text{in } z=\eta(x,t) \qquad \text{(Bernoulli)},$$

$$\frac{\partial\Phi}{\partial z} = \frac{\partial\eta}{\partial t} \qquad \text{in } z=\eta(x,t) \qquad \text{(Surface kinematic)}, \tag{F.15}$$

$$\frac{\partial\Phi}{\partial z} = 0 \qquad \text{in } z=-h \qquad \text{(Bottom kinematic)}. \tag{F.16}$$

Note that the domain Ω_z depends on the free surface η.

Using variable separation, the solution of the Laplace equation (F.14) can be written as $\Phi(x,z,t) = X(x)\,Z(z)\,T(t)$. Substituting in the problem and assuming $T(t) \neq 0\ \forall t \in \mathbb{R}^+$,

$$\frac{1}{X}\frac{d^2X}{dx^2} = -\frac{1}{Z}\frac{d^2Z}{dz^2} = -k^2,$$

where k is a constant. From the last equation, two ODE are derived for X and Z,

$$\frac{d^2X}{dx^2} + k^2X = 0,$$
$$\frac{d^2Z}{dz^2} - k^2Z = 0,$$

with solutions

$$X(x) = A\cos(kx) + B\sin(kx), \tag{F.17}$$
$$Z(z) = Ce^{kz} + De^{-kz},$$

where A, B, C, D are integration constants. Note that k is seen to be the *wavenumber* because the spatial structure of the solution is composed by $\cos(kx)$ and $\sin(kx)$. Hence, the potential has the form

$$\Phi(x,z,t) = (A\cos(kx) + B\sin(kx))\left(Ce^{kz} + De^{-kz}\right)T(t).$$

Using the kinematic bottom condition (F.16)

$$\left.\frac{\partial\Phi}{\partial z}\right|_{z=-h} = XTk\left(Ce^{-kh} - De^{kh}\right) = 0 \quad \Longrightarrow \quad Ce^{-kh} = De^{kh} = \frac{\Gamma}{2},$$

where Γ is again a constant which allows to write the potential in the form $\Phi(x,z,t) = \Gamma\cosh\left[k\left(z+h\right)\right](A\cos(kx) + B\sin(kx))\,T(t)$. Applying now the kinematic surface condition (F.15) results

$$\left.\frac{\partial\eta}{\partial t}\right|_{z=\eta} = k\Gamma\sinh(k\eta + kh)\,(A\cos(kx) + B\sin(kx))\,T(t),$$

that is an equation for the free surface that is still non-linear. Thus, the domain must be linearized to obtain a completely linear problem. The function $\sinh(k\eta + kh)$ can be approximated by $\sinh(kh)$ if the following conditions are both fulfilled:

$$\sinh(k\eta + kh) \simeq \sinh(kh) \quad \Longleftrightarrow \quad |\eta| \ll h \quad \text{and} \quad k|\eta| \ll 1. \tag{F.18}$$

The linearization of (F.18) is equivalent to evaluate the surface condition in $z = \eta = 0$. Moreover, it must hold independently from the bottom depth, that is, for shallow and deep water.

In shallow water, the wavelength is much greater than the depth of the bottom, that is $\ell \gg h$. Given the condition $|\eta| \leq h$, results that $\ell \gg |\eta|$, that is $k|\eta| \ll 1$ is always verified. Thus, the additional restriction $|\eta| \ll h$ must be added for shallow water. In deep water is just the opposite, so that $h \gg \ell$. Remembering the condition $\ell \geq H$, results that $|\eta| \ll h$ is always verified. The additional constrain in this case is $k|\eta| \ll 1$.

Summarizing, the condition (F.18) lead to the following two constraints,

$$|\eta| \ll h \quad \Longrightarrow \quad \frac{H}{D} \ll 1 \qquad \text{for shallow water,} \tag{F.19}$$

$$k|\eta| \ll 1 \quad \Longrightarrow \quad \frac{H}{\ell} \ll 1 \qquad \text{for deep water.} \tag{F.20}$$

The condition (F.19) implies small amplitude in comparison with the depth of the seabed, while (F.20) means small amplitude respect to the wavelength. Both are restrictions on the size of the waves arising from fully linearized equations.

F.3.2 Linear wave problem

Assuming the restrictions (F.19) and (F.20), the complete linearization of (F.13) gives

$$\left.\begin{aligned} \Delta_2\Phi + \frac{\partial^2\Phi}{\partial z^2} = 0 \qquad & \text{in} \quad -h(x,y) < z < 0, \\ \frac{\partial\Phi}{\partial z} = -\frac{1}{g}\frac{\partial^2\Phi}{\partial t^2} \qquad & \text{in} \quad z = 0, \\ \frac{\partial\Phi}{\partial z} = -\boldsymbol{\nabla}_2\Phi\cdot\boldsymbol{\nabla}_2 h \qquad & \text{in} \quad z = -h(x,y), \end{aligned}\right\} \tag{F.21}$$

which is from now on the problem of linear waves, which allows to express the free surface using the linearization of (F.11), that is

$$\eta(x,y,t) = -\frac{1}{g}\left.\frac{\partial\Phi}{\partial t}\right|_{z=0}. \tag{F.22}$$

F.4 Reduction to a two-dimensional problem

The propagation of sea waves is a purely two-dimensional problem, that is, the direction of propagation always lays on the $x-y$ plane. However, the coupling with the sea bottom imposes that the z coordinate must also be taken into account. In this section is introduced the mathematical procedure that allows to decouple the problem of linear waves from this z-dependence, reducing it to a completely two-dimensional problem.

F.4.1 Profile function

Here, the objective is to obtain a problem in the only variable z (profile problem). To this aim, the potential is written as

$$\Phi(x,y,z,t) = f(z)\,\phi(x,y,t), \tag{F.23}$$

where ϕ is now a reduced potential only depending on the propagation plane and the time, and $f(z)$ is the so called *profile function*. Hence, an analytical expression for

f is seek, complying with the problem (F.21). Thus, substituting (F.23) in (F.21), results

$$f\Delta_2\phi + f''\phi = 0 \qquad \text{in} \quad -h(x,y) < z < 0, \tag{F.24}$$

$$\phi f' = -\frac{f}{g}\frac{\partial^2\phi}{\partial t^2} \qquad \text{in} \quad z = 0, \tag{F.25}$$

$$\phi f' = -f\boldsymbol{\nabla}_2\phi\cdot\boldsymbol{\nabla}_2 h \qquad \text{in} \quad z = -h(x,y). \tag{F.26}$$

In order to eliminate the time dependency in (F.25), harmonic waves of frequency ω are considered. Hence, due to the linearity of the problem, the physical solution will be a linear combination of all the harmonic solutions. The reduced potential is therefore written as a monochromatic wave, that is

$$\begin{aligned}\phi(x,y,t) &= \phi_1(x,y)\cos(\omega t) + \phi_2(x,y)\sin(\omega t)\\ &= \mathrm{Re}\left[(\phi_1 + i\phi_2)e^{-i\omega t}\right] = \mathrm{Re}\left[\varphi(x,y)e^{-i\omega t}\right].\end{aligned} \tag{F.27}$$

Using the complex variable $\varphi(x,y)$, the reduced potential ϕ and the potential Φ are expressed in complex form as $\tilde{\phi}(x,y,t) = \varphi(x,y)e^{-i\omega t}$ and $\tilde{\Phi}(x,y,z,t) = f(z)\,\tilde{\phi}(x,y,t)$ to have,

$$\Phi = \mathrm{Re}\,\tilde{\Phi}, \quad \phi = \mathrm{Re}\,\tilde{\phi}.$$

Applying the dynamic surface condition (F.25) to the complex potential $\tilde{\phi}$ results

$$f' - \frac{\omega^2}{g}f = 0 \quad \text{in} \quad z = 0. \tag{F.28}$$

Similarly Equations (F.24) and (F.26) also should be rewritten using the complex potential. Nevertheless the frequency analysis does not cancel the spatial dependence on $(x,\, y)$ from these expressions. For the kinematic condition (F.26), this dependence is eliminated using the hypothesis of horizontal bottom, so that the gradient of the bottom function is eliminated, $\boldsymbol{\nabla}_2 h = \mathbf{0}$. Thus, the kinematic condition becomes

$$f' = 0 \quad \text{in} \quad z = -h. \tag{F.29}$$

For the Laplace equation (F.24), no additional hypothesis is needed, and the spatial dependence disappear writing it as

$$-\frac{\Delta_2\tilde{\phi}}{\tilde{\phi}} = -\frac{\Delta_2\varphi}{\varphi} = \frac{f''}{f} = \lambda \quad \Longrightarrow \quad f'' - \lambda f = 0 \quad \text{in} \quad -h < z < 0, \tag{F.30}$$

where λ is a constant.

F.4.2 Sturm-Liouville problem

Equation (F.30) with the boundary conditions (F.29) and (F.28) results in the following Sturm-Liouville problem, whose solution is the profile function,

$$\left.\begin{aligned} f'' - \lambda f = 0 \qquad & \text{in} \quad -h < z < 0, \\ f' - \frac{\omega^2}{g} f = 0 \qquad & \text{in} \quad z = 0, \\ f' = 0 \qquad & \text{in} \quad z = -h. \end{aligned}\right\} \tag{F.31}$$

Note that (F.31) is an homogeneous problem, hence functions $f(z)$ is not unique but belongs to a family of *eigenfunctions* associated to the different eigenvalues λ leading to a non-trivial solution ($f \neq 0$). These eigenvalues verify the relation $k^2 = \lambda$, being k the root of the characteristic polynomial, generating hence eigenfunctions with the form $f(z) = Ae^{\sqrt{\lambda}z} + Be^{-\sqrt{\lambda}z}$, where A and B are integration constants. Similarly to Section F.3.1, the kinematic bottom condition (F.29) is used to write this family as terms of an hyperbolic function,

$$f(z) = \Gamma \cosh\left[\sqrt{\lambda}(z+h)\right] = \Gamma \cosh[k(z+h)], \tag{F.32}$$

where Γ again is an integration constant. Note that $f(z)$ has a different value for each eigenvalue λ. Since (F.31) is not fully defined, Γ is not determined. Applying the boundary condition (F.28), a relation is obtained involving the parameter k, known as the *dispersion relation*,

$$kg \tanh(kh) = \omega^2, \tag{F.33}$$

where g is the gravity acceleration. Equation (F.33) shows the dependence of the wavenumber k, and therefore the wave velocity $c = \omega/k$, from the bottom depth h.

So far no restriction has been considered on the eigenvalue λ. A detailed analysis (see Dingemans (1997)) must be carried out for the cases $\lambda > 0$ ($k \in \mathbb{R}$) and $\lambda < 0$ ($k \in \mathbb{C}$) with objective of quantifying the number of eigenvalues providing a nontrivial solution to (F.31). It is worth mentioning that such analysis is focused exclusively on finding how many wavenumbers verify the dispersion equation.

For the real case $k \in \mathbb{R}$, Equation (F.33) can be written in the equivalent form using the notation $\Omega^2 = h\omega^2/g$ and $K = kh$ (Ω, $K \in \mathbb{R}$) as

$$\frac{\Omega^2}{K} = \tanh K.$$

The value of K verifying this new equation is given by the intersections of the curves Ω^2/K and $\tanh K$. As shown in Figure F.5, there is only one eigenvalue satisfying the Sturm-Liouville problem and producing a positive wavenumber. Denoting by $K_0 = k_0 h$ this unique root, the corresponding profile function or eigenfunction of (F.32), normally taken such that $f(0) = 1$ to have $\Gamma = 1/\cosh(kh)$, is

$$f_0(z) = \frac{\cosh[k_0(z+h)]}{\cosh(k_0 h)}. \tag{F.34}$$

For the complex case $k \in \mathbb{C}$, similarly, the dispersion relation is defined where now K is a pure complex number, say $K = i\kappa$ ($\kappa \in \mathbb{R}$). Using then the trigonometric relation $\tanh z = -i \tan(iz)$ $\forall z \in \mathbb{C}$, Equation (F.33) results

$$\frac{\Omega^2}{\kappa} = -\tan \kappa.$$

In Figure F.6 is shown the dispersion relation as function of κ. Now, differently to the previous case, there are infinite eigenvalues satisfying the Sturm-Liouville problem, with complex wavenumbers $k_n = K_n/h = i\kappa_n/h$. Similarly to (F.34), the eigenfunc-

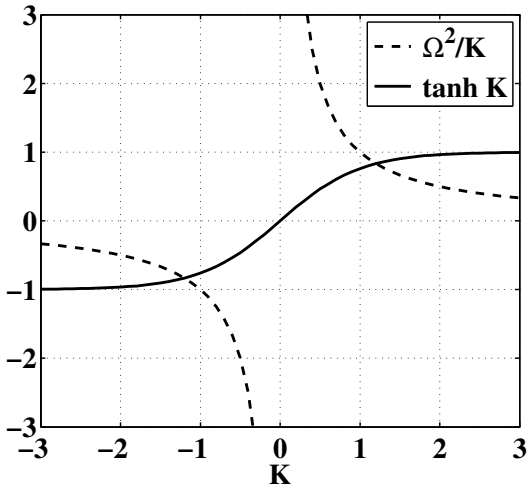


Figure F.5: Solution of the dispersion relation for the real value $K = kh$

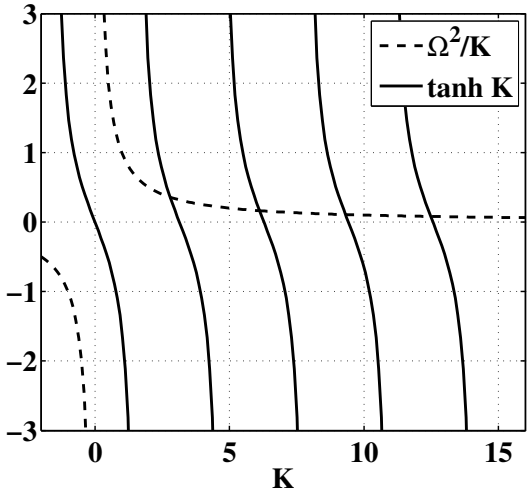


Figure F.6: Solution of the dispersion relation for the imaginary value $K = i\kappa$

tions associated to these eigenvalues are expressed as

$$f_n(z) = \frac{\cosh[k_n(z+h)]}{\cosh(k_n h)}, \tag{F.35}$$

where $n \in \mathbb{N}^*$ refers to the eigenvalue and the corresponding eigenfunction. Moreover, these eigenfunctions verify the orthogonality property in the profile,

$$\int_{-h}^{0} f_m(z) f_n(z)\, dz = 0 \quad \forall\, m, n \in \mathbb{N}^* \mid m \neq n.$$

Due to the linearity of the problem (F.31), the general form for the profile function can be written as a linear combination of the solutions (F.34) and (F.35), that is

$$f(z) = f_0(z) + \sum_{n=1}^{\infty} f_n(z) = \frac{\cosh[k_0(z+h)]}{\cosh(k_0 h)} + \sum_{n=1}^{\infty} \frac{\cosh[k_n(z+h)]}{\cosh(k_n h)}, \tag{F.36}$$

hence the complex potential $\tilde{\Phi}$ results

$$\tilde{\Phi}(x,y,z,t) = f(z)\,\tilde{\phi}(x,y,t) = f(z)\,\varphi(x,y)e^{-i\omega t} = \left(f_0\varphi_0 + \sum_{n=1}^{\infty} f_n\varphi_n\right)e^{-i\omega t}.$$

For the unidirectional case (see the example *long-crested waves* of Section F.3.1) and constant time $t_0 = 0$, results $\tilde{\Phi}(x,z) = f_0(z)\varphi_0(x) + \sum_{n=1}^{\infty} f_n(z)\varphi_n(x)$. In this case, $\varphi(x)$ is the solution of the Helmholtz equation

$$\frac{\partial^2 \varphi}{\partial x^2} + k^2\varphi = 0, \tag{F.37}$$

whose solution can expressed as (F.17), producing the following harmonic potential

$$\begin{aligned}
\tilde{\Phi}(x,z) &= f_0(z)[A_0\cos(k_0 x) + B_0\sin(k_0 x)] \\
&\quad + \sum_{n=1}^{\infty} f_n(z)[A_n\cos(k_n x) + B_n\sin(k_n x)] \\
&= a_0 f_0(z)e^{ik_0 x} + \sum_{n=1}^{\infty} b_n f_n(z)e^{ik_n x} \\
&= a_0 f_0(z)e^{ik_0 x} + \sum_{n=1}^{\infty} b_n f_n(z)e^{-\frac{\kappa_n}{h}x},
\end{aligned} \tag{F.38}$$

where $a_0 = A_0 + iB_0$ and $b_n = A_n + iB_n$, similarly to (F.27). Hence the real part of the expression (F.38) is the analytical solution of the linear wave propagation problem (F.21), particularized for a monochromatic wave traveling in direction x, at initial time t_0 and with constant bottom depth h.

At this point, a new hypothesis is introduced in order to simplify the expression (F.36) of the profile function. Note that the summing term corresponding to complex

wave numbers k_n in (F.38) presents an exponential with factor $-\kappa_n x/h$. The number κ_n, always positive, increases such that $\kappa_i < \kappa_j \ \forall\, i < j$, see Figure F.6. Hence, the exponential term reduces the contribution of the complex case $k \in \mathbb{C}$ to the wave propagation. The reduction is exponential in the propagation distance x, thus the eigenvalues associated to k_n are called *evanescent modes* as the wave presents the maximum amplitude in a point and then exponentially decays. Higher modes ($n \gg 0$) are more rapidly decaying.

Physically, an evanescent mode can be thought as a wave that adds a sudden variation to the free surface. This variation can also be interpreted as a sudden variation in the velocity of the wave train, generated by large gradients in the bottom bathymetry. The fundamental hypothesis of the Mild Slope equation is that the bathymetry is slowly varying. Hence, in the profile function the terms associated to the evanescent modes are eliminated and only the propagation of *progressive waves* is considered. [3] Thus, the complex potential has the form

$$\tilde{\Phi}(x, y, z, t) = \frac{\cosh[k(z+h)]}{\cosh(kh)} \varphi(x, y) e^{-i\omega t}, \tag{F.39}$$

where for notation is taken $k_0 \equiv k$. Dingemans (1997) describes the development of the Mild Slope equation for sudden variation of the bathymetry, thus taking into account the evanescent modes. From (F.39) can be derived the expression of the free surface (F.22) for a monochromatic progressive wave in complex form

$$\tilde{\eta}(x, y, t) = \frac{i\omega}{g} \varphi(x, y) e^{-i\omega t}, \tag{F.40}$$

such that $\eta = \mathrm{Re}\tilde{\eta}$.

Remark 8. *The profile function* (F.36) *is written with unitary coefficients of the linear combination. More generally, this linear combination can be written as* $f(z) = \beta_0 f_0(z) + \sum_{n=1}^{\infty} \beta_n f_n(z)$ *where* β_0 *and* β_n *are generic values. Taking out the evanescent modes,* β_0 *can be chosen in such a way that some condition on the free surface* (F.40)

[3] In Equation (F.38) this is equivalent to consider only the terms with the potential $\varphi_0(x)$, that is the linear combination (F.17) which produces a traveling progressive wave without dissipation.

is fulfilled. Normally, at the initial time $t_0 = 0$ *the wave height is prescribed as* H_0*, thus*

$$H_0 = 2|\tilde{\eta}(x, y, t_0)| = 2\beta_0 \frac{\omega}{g} |\varphi(x, y)| = \beta_0 \frac{\omega}{g} H \quad \Longrightarrow \quad \beta_0 = \frac{H_0}{H} \frac{g}{\omega},$$

resulting in a complex free surface of the form $\tilde{\eta}(x, y, t) = i\frac{H_0}{H}\varphi(x, y)e^{-i\omega t}$.

F.5 Derivation of the Mild Slope equation

In this section is detailed how to obtain a two-dimensional equation that models the propagation considering a seabed that can be variable. Thus, this will allow to model the refraction of the waves traveling over an irregular bathymetry. The basic hypothesis imposed in this deduction is the so called *mild slope* condition, after which the equation is named: it is assumed that the bottom slope is small and its shape is smooth.

The expression of the Mild Slope equation is deduced in time and frequency formulation. However, choosing the profile function as in Section F.4 leads to a time formulation that is actually dependent on the ω chosen. In principle, this could prevent the ability to of the model to reproduce a set of waves with different frequencies. However, Dingemans (1997) shows that the time formulation is able to generate waves *almost-harmonic*, where small variations of ω and k are allowed.

F.5.1 Starting point

The common idea in all the derivations of the Mild Slope equation (derived by Berkhoff (1972) [4]) is to suppose that the vertical structure of the domain is taken into account with the profile function $f(z, h(x, y))$, derived for horizontal bottom in Section F.4.2,

$$f(z, h) = \frac{\cosh[k(z + h)]}{\cosh(kh)}, \tag{F.41}$$

[4]For this reason the Mild Slope equation is often referred as the Berkhoff equation.

through the variable wavenumber $k = k(h(x, y))$ that complies the dispersion relation (F.33). In this way, (F.41) is exactly verified in the whole domain for horizontal bottom. In case of sloping bottom, the variation of the profile function is neglected, which is equivalent to consider a local verification of (F.41) around a given point (x, y). Taking derivative respect to the plane coordinates,

$$\boldsymbol{\nabla}_2 f = \frac{\partial f}{\partial h}\boldsymbol{\nabla}_2 h + \frac{\partial f}{\partial k}\boldsymbol{\nabla}_2 k = \left(\frac{\partial f}{\partial h} + \mu\frac{\partial f}{\partial k}\right)\boldsymbol{\nabla}_2 h \quad \Longrightarrow \quad \boldsymbol{\nabla}_2 f \propto \boldsymbol{\nabla}_2 h,$$

where the expression of μ if found deriving the dispersion relation. Thus, the gradient of the profile function is proportional to $\boldsymbol{\nabla}_2 h$. Neglecting $\boldsymbol{\nabla}_2 f$ means assume small variation of the bottom, which is the basic assumption in the Mild Slope equation.

Theorem F.1 (Second Green identity)**.** *Let ψ, ν be two functions of class C^2 defined in the open domain Ω, and $\boldsymbol{n}$ the normal to the boundary $\partial\Omega$. Hence, the following identity holds*

$$\int_\Omega \psi\Delta\nu - \nu\Delta\psi \, d\Omega = \oint_{\partial\Omega} \psi\frac{\partial\nu}{\partial\boldsymbol{n}} - \nu\frac{\partial\psi}{\partial\boldsymbol{n}} \, dS.$$

The derivation of the Mild Slope equation can follow two different paths: with a variational principle based on the energy conservation (*Hamiltonian* formulation) or using a more abstract rationale based on the Green theorem F.1. In the following, the second procedure is explained, referring to Dingemans (1997) for a detailed description of the first approach.

The basic idea and the principal objective is to achieve a single equation that brings together systems (F.21) and (F.31), knowing the profile function (F.41), in such a way that it is formulated only in the propagation plan. To this aim, the Laplace equation (F.24) is integrated in terms of the absolute potential Φ, [5] such that

$$\begin{aligned} f\Delta_2\phi + f''\phi = 0 \quad &\xrightarrow{f\neq 0} \quad f\left(f\Delta_2\phi + f''\phi\right) = 0 \quad \Longrightarrow \quad f\Delta_2\Phi + f''\Phi = 0 \\ &\xrightarrow{\text{(F.5)}} \quad \int_{-h}^{0} \frac{\partial^2\Phi}{\partial z^2} f - f''\Phi \, dz = 0. \end{aligned} \tag{F.42}$$

[5] The linear system (F.21) that must be reduced to a single equation is formulated with the absolute potential Φ.

In order to obtain a relation between the integral form (F.42) and the boundary conditions in the profile, the second Green identity is applied (see Green theorem F.1 for $\psi = f$ and $\nu = \Phi$), obtaining

$$\underbrace{\int_{-h}^{0} \frac{\partial^2 \Phi}{\partial z^2} f - f''\Phi \, dz}_{\text{Laplace}} = \underbrace{\left(\frac{\partial \Phi}{\partial z} f - f'\Phi\right)\Big|_{z=0}}_{\text{dynamic b.c.}} - \underbrace{\left(\frac{\partial \Phi}{\partial z} f - f'\Phi\right)\Big|_{z=-h}}_{\text{kinematic b.c.}}. \tag{F.43}$$

Equation (F.43) presents the terms $\partial^2\Phi/\partial z^2$ and f'' that can be substituted using the corresponding Laplace equations of systems (F.21) and (F.31). Similarly, the terms $\partial\Phi/\partial z$ and f' of the boundary conditions can be substituted. Hence

$$\begin{aligned}
\int_{-h}^{0} \frac{\partial^2 \Phi}{\partial z^2} f - f''\Phi \, dz \quad &\xrightarrow{\text{(F.5) and (F.30)}} \quad -\int_{-h}^{0} f\Delta_2\Phi + k^2 f\Phi \, dz, \\
\left(\frac{\partial \Phi}{\partial z} f - f'\Phi\right)\Big|_{z=-h} \quad &\xrightarrow{\text{(F.9) and (F.29)}} \quad -\left(f\boldsymbol{\nabla}_2\Phi \cdot \boldsymbol{\nabla}_2 h\right)|_{z=-h}, \\
\left(\frac{\partial \Phi}{\partial z} f - f'\Phi\right)\Big|_{z=0} \quad &\xrightarrow{\text{(F.28)}} \quad \left(\frac{\partial \Phi}{\partial z} f - \frac{\omega^2}{g} f\Phi\right)\Big|_{z=0},
\end{aligned}$$

such that Equation (F.43) results

$$\int_{-h}^{0} f\Delta_2\Phi + k^2 f\Phi \, dz = -\left(\frac{\partial \Phi}{\partial z} f - \frac{\omega^2}{g} f\Phi\right)\Big|_{z=0} - \left(f\boldsymbol{\nabla}_2\Phi \cdot \boldsymbol{\nabla}_2 h\right)|_{z=-h}, \tag{F.44}$$

which incorporates the linear wave problem (F.21) and the Sturm-Liouville problem (F.31). The Mild Slope equation derives from (F.44), particularizing the term $\partial\Phi/\partial z$ for either the frequency or time-frequency formulation.

F.5.2 Mild Slope equation in frequency domain

Using the harmonic analysis for a monochromatic wave ($\Phi = \mathrm{Re}\,\tilde{\Phi}$), the dynamic condition of problem (F.21) as a function of the complex potential $\tilde{\Phi}$ is

$$\frac{\partial \tilde{\Phi}}{\partial z} = \frac{\omega^2}{g}\tilde{\Phi},$$

such that the free surface term ($z = 0$) in (F.44) is zero, thus

$$\int_{-h}^{0} f\Delta_2\tilde{\Phi} + k^2 f\tilde{\Phi}\, dz = -\left(f\boldsymbol{\nabla}_2\tilde{\Phi}\cdot\boldsymbol{\nabla}_2 h\right)\Big|_{z=-h}, \tag{F.45}$$

where $\tilde{\Phi}(x,y,z) = f(z,h(x,y))\,\varphi(x,y)$.

Theorem F.2 (Leibniz rule)**.** *Let $p(z,\alpha)$ be a function defined in $\Omega \times J$ with Ω compact and J open. Hence, if p is differentiable in J $\forall z \in \Omega$ and that $\partial p/\partial\alpha \in C^0(\Omega \times J)$, the following relation holds*

$$\frac{d}{d\alpha}\int_{a(\alpha)}^{b(\alpha)} p(z,\alpha)\, dz = \int_{a(\alpha)}^{b(\alpha)} \frac{\partial p(z,\alpha)}{\partial\alpha}\, dz + p(b,\alpha)b' - p(a,\alpha)a'.$$

The objective now is to reduce Equation (F.45) grouping together the integrals in the profile such that they only affect function $f(z)$. The integrals will be computed analytically as the expression for (F.41) is know. Rewriting the term $\Delta_2\tilde{\Phi}$ results

$$\begin{aligned}\int_{-h}^{0} f\Delta_2\tilde{\Phi}\, dz &= \int_{-h}^{0} f^2\Delta_2\varphi\, dz + \int_{-h}^{0} f\varphi\Delta_2 f\, dz + \int_{-h}^{0} 2f\boldsymbol{\nabla}_2\varphi\cdot\boldsymbol{\nabla}_2 f\, dz \\ &= \underbrace{\int_{-h}^{0} \boldsymbol{\nabla}_2\cdot(f^2\boldsymbol{\nabla}_2\varphi)dz}_{\{1\}} + \int_{-h}^{0} f\varphi\Delta_2 f\, dz,\end{aligned}$$

where $\{1\}$ is rewritten using the Leibniz rule (see Theorem F.2 for $\alpha = (x,y)$ and $p = f^2\boldsymbol{\nabla}_2\varphi$), that is

$$\int_{-h}^{0} \boldsymbol{\nabla}_2\cdot(f^2\boldsymbol{\nabla}_2\varphi)dz = \boldsymbol{\nabla}_2\cdot\int_{-h}^{0} f^2\, dz\boldsymbol{\nabla}_2\varphi - \left(f^2\boldsymbol{\nabla}_2\varphi\cdot\boldsymbol{\nabla}_2 h\right)\big|_{z=-h}.$$

Using this result and developing the expression for $\tilde{\Phi}$ and its gradient $\boldsymbol{\nabla}_2\tilde{\Phi} = \varphi\boldsymbol{\nabla}_2 f + f\boldsymbol{\nabla}_2\varphi$, Equation (F.45) results

$$\boldsymbol{\nabla}_2\cdot\int_{-h}^{0} f^2\, dz\boldsymbol{\nabla}_2\varphi + k^2\int_{-h}^{0} f^2\, dz\,\varphi = -\varphi\int_{-h}^{0} f\Delta_2 f\, dz - (f\varphi\boldsymbol{\nabla}_2 f\cdot\boldsymbol{\nabla}_2 h)|_{z=-h}. \tag{F.46}$$

The right hand side of (F.46) contains informations about the slope of the sea bottom, if the terms $\Delta_2 f$ and $\boldsymbol{\nabla}_2 f$ are computed as

$$\Delta_2 f = \frac{\partial^2 f}{\partial h^2}(\boldsymbol{\nabla}_2 h)^2 + \frac{\partial f}{\partial h}\Delta_2 h \quad \text{and} \quad \boldsymbol{\nabla}_2 f = \frac{\partial f}{\partial h}\boldsymbol{\nabla}_2 h.$$

The terms of the order $(\boldsymbol{\nabla}_2 h)^2$, associated to the slope, and the one of the order $\Delta_2 h$, associated to the smoothness, are negligible. Suh et al. (1997) describe a modification of the Mild Slope equation that allows rapid variations of the topography, hence where these terms are not neglected. Equation (F.46) is simplified as

$$\boldsymbol{\nabla}_2 \cdot \int_{-h}^{0} f^2\, dz \boldsymbol{\nabla}_2 \varphi + k^2 \int_{-h}^{0} f^2\, dz\, \varphi = 0.$$

The integration of the square of the profile function is

$$g \int_{-h}^{0} f^2\, dz = c c_g, \tag{F.47}$$

where $c = \omega/k$ is the phase celerity and $c_g = d\omega/dk$ is the so called *group celerity*, computed using the dispersion relation (F.33). Using this result, the Mild Slope equation in frequency domain is

$$\boldsymbol{\nabla}_2 \cdot c c_g \boldsymbol{\nabla}_2 \varphi + k^2 c c_g \varphi = 0. \tag{F.48}$$

F.5.3 The Mild Slope equation in time-frequency domain

Here the aim is to obtain an equation similar to (F.48) but with the ability to model the temporal evolution of the wave train. Starting again form Equation (F.44), the dynamic surface condition is replaced in the linear system (F.21), obtaining

$$\int_{-h}^{0} f\Delta_2 \Phi + k^2 f \Phi\, dz = \frac{1}{g}\left(\frac{\partial^2 \Phi}{\partial t^2} + \omega^2 \Phi\right)\Big|_{z=0} - \left(f \boldsymbol{\nabla}_2 \Phi \cdot \boldsymbol{\nabla}_2 h\right)|_{z=-h}, \tag{F.49}$$

where now, differently from (F.45), the surface term is not zero and the potential Φ is a function of time, $\Phi(x, y, z, t) = f(z, h(x, y))\,\phi(x, y, t)$. Using the same mathematical manipulation as in Section F.5.2, results

$$\boldsymbol{\nabla}_2 \cdot \int_{-h}^{0} f^2\, dz \boldsymbol{\nabla}_2 \phi + k^2 \int_{-h}^{0} f^2\, dz\, \phi = \frac{1}{g} \underbrace{\left(\frac{\partial^2 \phi}{\partial t^2} + \omega^2 \phi \right) \Big|_{z=0}}_{\{1\}},$$

where the profile function has been particularized to have $f(0, h) = 1$. Note that $\{1\}$, evaluated in the surface, can be incorporated in a natural way in the equation as it is formulated in the plane $z = 0$. Applying again the definition (F.47) and reordering, the Mild Slope equation in time-frequency domain results

$$\frac{\partial^2 \phi}{\partial t^2} - \boldsymbol{\nabla}_2 \cdot cc_g \boldsymbol{\nabla}_2 \phi + (\omega^2 - k^2 cc_g)\phi = 0, \tag{F.50}$$

where in this case $\phi = \phi(x, y, t)$.